FERDINAND PORSCHE
FERN FH

Agile Softwareprojekte und deren Erfolgsfaktoren

Bachelorarbeit

eingereicht von: Sebastian Hautz
Matrikelnummer: 52105581

im Fachhochschul-Bachelorstudiengang Wirtschaftsinformatik (0470)
der Ferdinand Porsche FernFH

zur Erlangung des akademischen Grades <einer/eines>

Bachelor of Arts in Business

Betreuung und Beurteilung: DI Eszter Geresics-Földi MSc BSc

Wiener Neustadt, 04.2024

Ehrenwörtliche Erklärung

Ich versichere hiermit,

dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.

dass ich diese Bachelorarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.

Wien Hautz, Sebastian 19.04.2024

Unterschrift

Creative Commons Lizenz

Das Urheberrecht der vorliegenden Arbeit liegt bei <der Autorin/beim Autor>. Sofern nicht anders angegeben, sind die Inhalte unter einer Creative Commons <„Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz“ (CC BY-NC-SA 4.0)> lizenziert.
Die Rechte an zitierten Abbildungen liegen bei den in der jeweiligen Quellenangabe genannten Urheber*innen.

Die Kapitel <2 bis 9> der vorliegenden Bachelorarbeit wurden im Rahmen der Lehrveranstaltung „Bachelor Seminar 1“ eingereicht und am 01.12.2023 als Bachelorarbeit 1 angenommen.

Kurzzusammenfassung: Agiles Softwareprojekte und deren Erfolgsfaktoren

Agile Softwareentwicklung, ist seit einigen Jahren der Defacto Standard, wenn es um Softwareentwicklung geht, allerdings ist der Erfolg dieser Projekte von vielen unterschiedlichen Faktoren abhängig.

Diese Arbeit untersucht, mithilfe einer Umfrage und von Interviews betroffener Personen, die in der agilen Softwareentwicklung arbeiten, soll evaluiert werden, welche drei Faktoren, die häufig genannten sind und somit das Agile Arbeiten fördern. Hierzu wurde ein Fragebogen formuliert, welcher die gestellten Fragen beinhaltet sowie die enthaltenen Fragen übergeordneten Kategorien zugeteilt. Anhand dieser Einteilung lässt sich feststellen, welche der 4 angeführten Kategorien den Haupteinfluss laut Meinung der Befragten auf den Erfolg im agilen Umfeld hat.

Im Zuge dieser Arbeit wurden diese Faktoren untersucht, diese Faktoren wurden mithilfe einer Umfrage festgestellt. Während der Auswertung der Umfrage wurden diese Erfolgsfaktoren identifiziert, welche mit der von mir gestellten Hypothese verglichen wurden. Durch diesen Vergleich hat sich die von mir gestellte Hypothese als falsch herausgestellt.

Schlagwörter:

Agile, Softwareentwicklung, Organisation, Erfolgsfaktoren, Teamarbeit

Abstract: Agile Software projects and their success factors

Agile software development has been the de facto standard for software development for several years, but the success of these projects depends on many different factors.

This work examines, with the help of a survey and of interviews of persons concerned, who work in the agile software development, is to be evaluated, which three factors are the most frequently mentioned and promote thus the agile working. For this purpose, a questionnaire was formulated that contains the questions asked and assigns the questions contained to superordinate categories. Based on this classification, it can be determined which of the 4 categories listed has the main influence on success in the agile environment according to the opinion of the respondents.

Keywords:

Agile, Softwaredevelopment, Organization, Success factor, Teamwork

Inhaltsverzeichnis

1. EINLEITUNG 1

1.1 Ausgangssituation 1

1.2 Forschungsinteresse, Forschungsfrage und Hypothese 2

1.2.1 Forschungsfrage und Forschungsziel 2

1.2.2 Hypothese 3

1.3 Forschungsmethode 3

1.4 Abgrenzung 3

2. GRUNDLAGEN UND AKTUELLER STAND DER WISSENSCHAFT 5

2.1 Software-Lebenszyklus 6

2.1.1 Spezifikationsanalyse 8

2.1.2 Design 8

2.1.3 Implementierung 9

2.1.4 Testen 9

2.1.5 Release 10

2.1.6 Wartung 10

2.2 Agiles Manifest 10

2.2.1 Agilen Werte 11

2.2.2 Agilen Prinzipien 12

2.3 Vorgehensmodelle in der Softwareentwicklung 13

2.3.1 Wasserfall Modell 13

2.3.2 Scrum 15

2.3.3 Kanban 20

2.4 Potenzielle Stör- und Erfolgsfaktoren 22

2.4.1 Sektoren 22

2.4.2 Störfaktoren 23

2.4.3 Erfolgsfaktoren 25

3. GRUNDLAGEN AGILES ARBEITEN IM SOFTWAREPROJEKT 26

4. FORSCHUNGSMETHODE 31

4.1 Fragebogen und Interview-Leitfaden 31

4.1.1 Quantitative Umfrage 31

4.1.2 Mögliche andere Methode 32

4.1.3 Fragestellungen 32

4.1.4 Auswertungen 34

5. DURCHFÜHREN DER BEFRAGUNG UND INTERVIEWS 36

6. ANALYSE DER ERGEBNISSE 37

6.1 Auswertungsmethodik 37

6.2 Auswertungstool 37

6.3 Einleitungsfragen 37

6.4 Gesamtauswertung 39

6.5 Die drei meistgenannten Faktoren 40

6.5.1 Frage 11 Einfache und Verständliche Arbeitsprozesse 41

6.5.2 Frage 10 Kommunikationsfluss des Unternehmens 43

6.5.3 Frage 21 Definition of Ready, Definition of Done 45

6.6 Interessante Ergebnisse 47

6.6.1 Frage 8 Altersgruppe 47

6.6.2 Frage 9 Kultur 48

6.6.3 Frage 7 Sprachkenntnisse 49

6.6.4 Frage 15 Agile Methodik 50

6.6.5 Frage 13 Self Managing Teams oder Management durch das Unternehmen 51

6.6.6 Frage 24 ausführliche Dokumentation 52

7. BEANTWORTUNG DER FORSCHUNGSFRAGE UND EVALUIERUNG DER HYPOTHESE 53

7.1 Beantwortete Forschungsfrage 53

7.2 Evaluierung der Hypothese 53

8. ZUSAMMENFASSUNG UND AUSBLICK 55

9. FORMELN, FRAGENKATALOG UND ROHDATEN 57

9.1 Formeln 57

9.2 Fragenkatalog 57

9.3 Rohdaten 59

10. ABBILDUNGSVERZEICHNIS 60

11. LITERATURVERZEICHNIS 61

1. Einleitung

1.1 Ausgangssituation

Wir alle verwenden in unserem täglichen Leben Software, sei es auf einem Smartphone, Computer oder anderen Elektronischen Geräten, zuhause oder in der Arbeit, diese Software, die in unseren Täglichen Leben zur Verwendung kommt, muss allerdings zuerst entwickelt werden. Dies geschieht durch Softwareentwicklung, dieser Softwareentwicklungsprozess hat sich im Laufe der Jahre organisiert und standardisiert, es kamen dadurch immer mehr theoretische Ansätze hinzu, um diesen Entwicklungsprozess zu vereinheitlichen und zu Kategorisieren. Diese Kategorisierung beinhaltet eine Vielzahl unterschiedlich durchgeführter Schritte, allerdings kann der Ansatz, sowie die zu verwendende Schritte, leicht von Methode zu Methode unterschiedlich sein. (siehe dazu „*Kapitel Vorgehensmodelle in der Softwareentwicklung“*)

Der erste diese Ansätze ist die sogenannte „Wasserfall Methode“ (Burkhard 2012) hier wird ein Schritt nach dem anderen Sequenziell, durchgeführt und ein Schritt wird erst gestartet, wenn der vorgegangene Schritt abgeschlossen ist. Dies war initial ein Ansatz mit dem Jahrelang gearbeitet werden konnte, aber umso größer Firmen, Projekte, Code sowie deren Teams wurden, umso mehr wurden die negativen Aspekte dieser Methodik klar, so wuchs der Frust der betroffenen stätig.

2001 wurde dann ein neuer Weg gesucht, um eine alternative zu dem bestehenden Softwareentwicklungsprozess zu bieten, da die bereits existierenden Methoden in den Augen der Autoren des Manifests zu viele Probleme aufwerfen. Aus diesem Findungsprozess entstand das Agile Manifest, welches die Probleme der Klassischen Methode beheben oder beziehungsweise vermindern soll (Kent, et al. 2001), seit seiner Entstehung findet die Agile Softwareentwicklungsmethode immer mehr Anhänger, und somit auch mehr Verwendung im täglichen Arbeitsalltag, mehrerer Unternehmen und Entwicklungsteams. (Kent, et al. 2001) Durch den erhöhten Einsatz dieser neuen Methode, sind immer mehr Variationen der Agilen Methodik entstanden wie etwa Scrum, Kanban, SAFe und ähnliche, die unterschiedlichen Ansätze der Agilen Methodiken verwenden welche spezifischen Probleme im Softwareentwicklungsmethodik lösen sollen.

Allerdings finden sich trotz unterschiedlich verwendeter Agiler Methodik immer wieder Störfaktoren, welche von den Betroffenen nicht immer gleich erkannt werden, auch können diese Störfaktoren abhängig von den bewertenden Personen unterschiedlich gewichtet werden, welches die Priorisierung des auflösen dieser Störfaktoren erschwert. Dieses Ungleichgewicht in der Wahrnehmung und Priorisierung zur Lösung dieser Störfaktoren kann zu Unmut im Team führen,

welches Potenzielle Konfliktherde schürt, und schlimmstenfalls zu einem Scheitern der Teamarbeit oder gar eines Projektes führen kann.

Diese Faktoren können auch einen Positiven Ansatz haben, da allerdings auch hier die Wahrnehmung, sowie die Gewichtung dieser Faktoren, von jeder Person unterschiedlich Bewertet werden, auch diese Unterschiedliche Wahrnehmung der Faktoren kann zu Unmut und Unverständnis anderer Teammitglieder führen, aus diesem Grunde ist das Identifizieren Positiver Einfluss Faktoren genauso maßgeblich für den Erfolg von Agilen Softwareentwicklungsprojekten wichtig, wie das Identifizieren von Störfaktoren, der Fokus dieser Arbeit liegt allerdings auf dem Identifizieren von Erfolgsfaktoren. Da aber wie bereits beschrieben, diese Wahrnehmung unterschiedlich ausfällt, gilt es diese Positiven generellen Faktoren welche unabhängig von Projekt und Agiler Methodik zu identifizieren, und zu gewichten.

1.2 Forschungsinteresse, Forschungsfrage und Hypothese

Der Grund, wieso ich mich für dieses Thema interessiere, ist folgender, 2019 Absolvierte ich die Abend HTL-Ottakring mit den Schwerpunkt Informatik, und habe währenddessen und nach dem Absolvieren der Abendschule in Unterschiedlichen IT-Softwarefirmen gearbeitet, welche auch unterschiedliche Arten der Softwareentwicklungsmethodik verwendet haben (Scrum, Kanban oder Ähnliches). Allerdings stieß ich öfter auf oben angeführte Problematik der Unterschiedlichen Wahrnehmung von Faktoren, welche den Arbeitsprozess positiv beeinflussen sollten, dies ist gerade in den ersten Phasen einer Teamfindung (Tuckman 1965) und Entwicklung eines Teams deutlich problematischer, da die Teamfindung erst stattfinden muss.

Aus diesem Grunde möchte ich Untersuchen und feststellen, ob gewisse Punkte unabhängig von Firmen Umfeld, sowie auch unabhängig von der eingesetzten Softwareentwicklungsmethodik sind. Diese identifizierten Faktoren sollen übergeordnete Kategorien zugewiesen werden, so soll es möglich sein einen Schwerpunkt beziehungsweise Kernpunkt zu identifizieren, welche unabhängig von Projektumfeld, sowie Agiler Methodik sind, um in Zukünftigen Projekten und Teams eine bessere Arbeitsweise und Arbeitsklima zu erzielen, sowie Potenzielle Konfliktpunkte zu eliminieren, da sich einige allgemein gültige Faktoren Identifizieren lassen.

1.2.1 Forschungsfrage und Forschungsziel

Die Entwicklung der Forschungsfrage ist entstanden durch die Problematik des Zusammenarbeitens in Team unabhängig von Firmenumfeld sowie der Verwendung des verwendeten Vorgehensmodelles, die Frage, mit der sich diese Arbeit beschäftigt ist, nun:

„Welche sind die 3 meistgenannten Erfolgsfaktoren für das agile Arbeiten in Softwareprojekten? „

Und Somit ist das Ziel dieser Arbeit, das Identifizieren der 3 Meistgenannten Erfolgsfaktoren, für das agile Abreiten in Softwareprojekten, dies soll erreicht werden durch eine gestellte Umfrage, und ein Leitfaden-gestütztes Interview, welche anhand einer Skala bewertet werden. Durch diese Skala lässt sich jeder genannte Faktor gewichten, durch diese Gewichtung lassen sich dann die drei meistgenannten Faktoren erkennen und Identifizieren. Hierbei soll allerdings nicht auf ein Spezifischen Projekt beziehungsweise eine Agile Methodik eingegangen werden (Scrum, SAFe), es soll lediglich helfen eine allgemeine Aussage zu treffen.

1.2.2 Hypothese

Die Hypothese aufgrund der Obigen Forschungsfrage dieser Arbeit ist:

“Die 3 meistgenannten Erfolgsfaktoren des agilen Arbeitens sind:

Klare Definition des Projektes/Scopes

Ein klarer Kommunikationsfluss innerhalb des Unternehmens

Einfache und Verständliche Arbeitsprozesse“

Diese genannte Hypothese stützt sich auf gesammelter Erfahrung in den Jahren meiner Karriere, und Unterschiedlicher Tätigkeiten in der Branche der Softwareentwicklung.

1.3 Forschungsmethode

Nach einer ausführlichen Literatur Recherche, wurde für die Beantwortung der Hypothese die Methodik, der Befragung gewählt, sowie der Leitfaden gestützte Interview. Eine ausführliche Beschreibung, wieso diese Methode gewählt wurde, ist dem Kapitel „*Forschungsmethode*“ zu entnehmen.

1.4 Abgrenzung

Das Ziel dieser Arbeit ist nicht das Vergleichen, von mehreren Vorgehensmodellen, oder Softwareentwicklungspraktiken zueinander, auch nicht der Vergleich von Klassischer und Agilen Softwareentwicklungsmethodik. Auch werden hier nicht die Vorteile sowie die Nachteile zwischen diesen Unterschiedlichen Vorgehensmodellen aufgelistet, beziehungsweise diese Verglichen. Vernachlässigt wird sowohl die Größe von Unternehmen, in denen die Befragten tätig sind, sowie die zugehörige Branche des Unternehmens, da dies für das Ziel und Ergebnis dieser Arbeit vernachlässigbar ist beziehungsweise nicht relevant. Ebenfalls ist die verwendete Implementierungssprache der Befragten Person nicht relevant. Natürlich spielen diese oben genannten Faktoren im täglichen Leben eine Rolle und sind sehr wohl für den Erfolg oder

Misserfolg eines Projektes beziehungsweise Teams relevant. Ebenfalls vernachlässigt werden Störfaktoren im Detail diese werden zwar oberflächlich erwähnt in Kapitel „*Potenzielle Stör- und Erfolgsfaktoren*“ doch werden diese nicht untersucht, sondern lediglich betrachtet für ein generelles Verständnis für diese Arbeit.

2. Grundlagen und Aktueller Stand der Wissenschaft

In diesem Kapitel möchte ich nun auf generelle Begrifflichkeiten der Softwareentwicklung, sowie auch auf die Begrifflichkeiten der Umfrage Methodiken eingehen, sowie den Aktuellen Stand der Wissenschaft wiedergeben. Welche für das weitere Verständnis dieser Arbeit wichtig sind. Da diese Begriffe häufiger Verwendungen finden, und es auch deshalb von großer Wichtigkeit ist das hier ein allgemeines Verständnis vorliegt, beziehungsweise ein gewisses Grundverständnis aufgebaut wird, da es sonst zu Unklarheiten im weiteren Verlauf kommen kann. Diesem, soll mit dem folgenden Kapitel vorgebeugt werden. Aufgrund des Umfanges dieses Themas kann leider nicht auf alle Aspekte und Details, welche hier behandelt werden, eingegangen werden, da dieses Thema zu Umfangreich ist, und somit den Rahmen dieser Arbeit sprengen würde. Es soll lediglich ein Grundwissen aufgebaut werden sowie den Aktuellen Stand der Wissenschaft wiedergeben, der zum Zeitpunkt des Schreibens dieser Arbeit gültig war, beziehungsweise dem Aktuellen Stand entsprach. Allerdings ist es wichtig die Entstehungsgeschichte der Softwareentwicklung, sowie die Geschichte einer der wichtigsten Vorgehensmodelle zu erläutern, da dies für das weitere Verständnis wichtig ist, da durch das Durcharbeiten der Kapitel das Historische Wachstum dieser Methodiken betrachtet werden kann.

Auch versucht dieses Kapitel, den Aktuellen Stand der Wissenschaft, in Bezug auf die wichtigen Themen für diese Arbeit, darzustellen und wieder zu geben, da gerade in der heutigen Zeit Agile Softwareentwicklung so Präsent ist wie noch nie zuvor, in Regelmäßigen Abständen entstehen neue Vorgehensmodelle, und Best-Use Praktiken für eine bereits existierende Agile Methodik, oder eine gar neue Methodik wird entwickelt, welche eine andere Problematik lösen soll oder welche eine bereits bestehende und bekannte Problematik eliminieren soll. Die von uns verwendete Software, wird von Jahr zu Jahr größer, so besteht Google mit Stand Juli 2016, aus etwa 2 Milliarden Zeilen Source-Code, 9 Millionen Source Dateien, und einer Gesamtgröße von etwa 86TB (Potvin and Levenberg 2016), und Google ist bei weitem nicht die einzige Firma, die

ein erhöhtes Datenaufkommen vorzuweisen hat, allerdings wird auch der

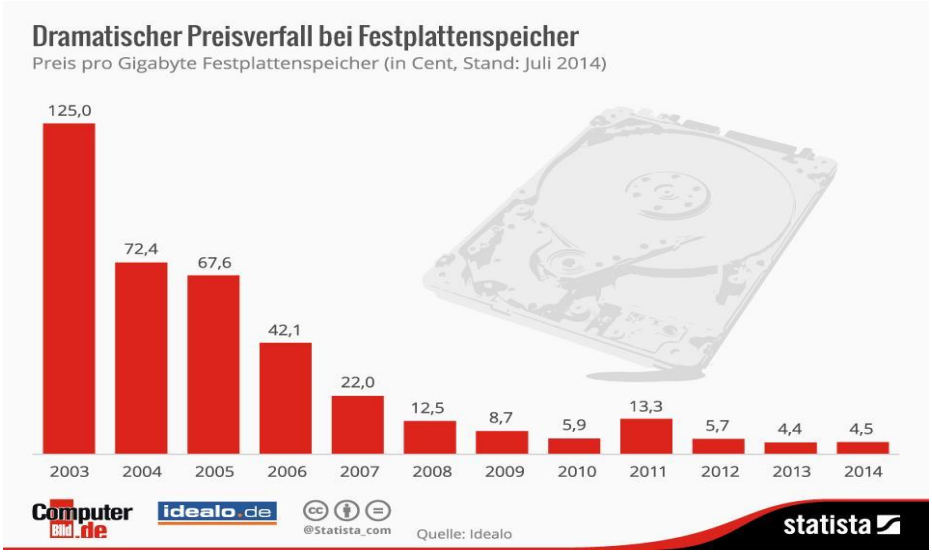


Abbildung 1 Festplatten Preis Pro GB Entwicklung (Brandt 2014)

benötigte Festplattenspeicher immer billiger, wie auf „*Abbildung 1 Festplatten*“ ersichtlich, dies erlaubt es auch, dass billig mehr zusätzlicher Speicher erworben werden kann.

Diese Entwicklung aus immer billiger werdendem Speicher, führt dazu das eine immer größer werdenden Maße, an Daten sich weltweit bewegen, und diesen Maßen gilt es zu kontrollieren, auch ermöglicht dies immer größere Daten von Code, welche wiederrum Diverse Agile Methodiken benötigten und Ansätze diese Giganten Softwarelösungen zu managen.

2.1 Software-Lebenszyklus

Softwaresysteme durchlaufen bei ihrer Entwicklung und Verwendung, einen ständig wiederkehrenden Prozess, den sogenannten Software-Lebenszyklus, hier handelt es sich um eine Reihe von Schritten, welche eine Software im Laufe ihres Lebens durchläuft. Es ist allerdings kein Vorgehensmodell im klassischen Sinne, sondern, hierbei handelt es sich um eine wiederkehrende Abfolge von notwendigen Prozessen, den dieses Softwaresystem in seiner Entstehung, und Lebensdauer, durchläuft.
Die genaue Anzahl der Schritte, sowie deren Bezeichnung, können abhängig von der Verwendeten Literatur variieren. Allerdings finden sich in den Grundzügen immer dieselben Schritte, sowie die damit verbundenen Notwendigen beziehungsweise durchgeführten Aktionen, die durchlaufen werden müssen, dass eine Kontinuierliche Verbesserung der Software stattfindet, und somit die entwickelte Software so lange wie möglich verwendet werden kann, ohne dass eine vollständige neue Implementierung notwendig ist.

Abbildung 2 Software-Lebenszyklus (eigene Erstellung)

Die oben abgebildete Darstellung, des Software-Lebenszyklus ist eine Abwandlung, aus dem Studienheft der Ferdinand Porsche FernFH (Schikuta, Derntl and Wanek 2019), und somit beziehe ich mich auch auf die dort angeführten Schritte, denn wie bereits im vorhergehenden Absatz beschrieben, sind die durchgeführten Schritte, abhängig von der Verwendeten Literatur, allerdings sind sie trotzdem in ihren Grundzügen der Verwendung gleich.

Wie auf dem Bild ersichtlich, ist der Lebenszyklus einer Software ein in sich geschlossener Kreis, da Softwareentwicklung ein nie endend wollender Prozess ist, da dieser Kreislauf immer wieder erneut durchlaufen wird, dieser Vorgang des immer wiederkehrenden durchlaufen desselben Prozesses nennt sich Iteration. Iteration bedeutet, aus dem Lateinischen in das Englische übersetzt „*again*“ welches, wenn weiter in das Deutsche übersetzt, „*wieder*“ bedeutet, somit steht es in einer Sinngemäßen Verwendung für die Bezeichnung des mehrfachen Durchlaufens, (Vocabulary.com´Dictionary n.d.) von Software oder in diesem Falle das mehrfache Wiederholen derselben aufeinander aufeinanderfolgenden Schritten, welche wiederrum für eine neuerliche Iteration notwendig sind.
Somit ist dies ein Iterativer Prozess, ein Prozess, welcher immer wieder durchlaufen wird und mit

jeder Iteration Neuerungen und Veränderung mit sich bringt. Dieser Softwarelebenszyklus ist in einzelne Schritte unterteilt, nach durchlaufen aller Schritte entsteht am Ende eine funktionierende Software, und mit jeder Iteration wird diese Software verändert, beziehungsweise auf den vorhergehenden Stand aufgebaut.

(Balzert 2011) (Burkhard 2012)

2.1.1 Spezifikationsanalyse

Hier werden die Funktionalen Anforderungen sowie die nicht- Funktionalen formuliert, dies erfolgt meistens mit Unterschiedlichen Analyse Methodiken, anhand dieser Analyse, wird dann eine Modellierungssprache (zum Beispiel UML) verwendet, um Domainspezifische Begriffe, Konzepte und vorgaben, welche für die Anforderung wichtig sind festzuhalten.

Auch finden sich hier weitere Technische Details, wie die verwendete Implementierungssprache, Art der Datenbank, und ähnliches, mit der durch die Modellierungssprache umgesetzte Systemlandschaft, können auch die ersten Objektorientierten Ansätze getroffen werden, wie zum Beispiel die Anwendungsfälle sowie Verwendungszwecke, von übergeordneten Schnittstellen – und Datenklassen. Dieser Schritt ist auch im laufenden Betrieb immer wieder zu durchzuführen, wenn eine Notwendigkeit besteht, so zum Beispiel, wenn eine neue Funktion zugebaut werden soll, oder beziehungsweise geändert werden muss.

(Balzert 2011) (Burkhard 2012)

2.1.2 Design

Hier werden die vorher gesammelten Punkte, der Spezifikationsanalyse in eine Software-Repräsentation übersetzt, dies kann zum Beispiel erfolgen durch die Verwendung der Use-Case Notation geschehen. Allerdings gibt es mehrere Möglichkeiten der Darstellung, auch wird hier die Software-Architektur definiert also zum Beispiel ob bestimmte Design Patterns (zum Beispiel Singleton) verwendet, werden sollen.

(Balzert 2011) (Burkhard 2012)

Auch werden hier im Weiteren die Struktur des zu Implementierenden Systems festgelegt, die sogenannte Systemarchitektur, und in späteren Iterationen auch, wie das System sich in der Laufzeit zu verhalten hat, oder wie bestehende Implementierungen verändert werden müssen, um den neuen Anforderungen gerecht zu werden.

Dieser Schritt ist während der Verwendung eines Systems von großer Wichtigkeit, da mit jeder Iteration neue Anforderungen an das Softwaresystem gestellt werden oder bestehendes verhalten geändert werden muss.

(Balzert 2011) (Burkhard 2012)

2.1.3 Implementierung

Die zuvor gesammelten Anforderungen, werden in der Spezifizierten Implementierungssprache (zum Beispiel Java, C++) umgesetzt, dieser Schritt ist einer der Arbeitsintensivsten Schritte da die zuvor gesammelten Verbalen wünsche, in Code und Logik umgesetzt werden müssen. Auch wird hier die Umgebung aufgesetzt, oder beziehungsweise Notwendige System mit installiert und aufgesetzt, wie zum Beispiel eine Datenbank, die benötigt wird, da die zu Implementierende Software Operative Daten speichern muss. Im weiteren Lebens Zyklus und somit in einer der späteren Iterationen einer Software, ist die Adaptierung, sowie Neugestaltung, des bestehenden bereits Implementieren Code wichtig, auch darf der Aspekt des Bugfixings (Fehlerbehebung) nicht vernachlässigt werden.

(Balzert 2011) (Burkhard 2012)

2.1.4 Testen

Sind die Anforderungen ausimplementiert, müssen diese getestet werden, um eine Funktion des Systems zu gewährleisten, da das gelieferte Produkt natürlich eine möglichst geringe Anzahl an Fehlern enthalten soll. Die Schritte der Implementierung und des Testens gehen oft Hand in Hand, und laufen somit oft zeitgleich ab, der Schritt des Testens kann auch von einer anderen Person übernommen, welche die Implementierung durchgeführt hat, um so eine möglichst große Anzahl an Fehlern vorab zu finden.

Hier findet auch oft eine Automatisierung von den erstellten Testfällen statt, diese Sammlung von erstellten Testfällen, ist das sogenannte Regressionsportfolio, dank diesem Regressionsportfolios lassen sich in Regelmäßigen Abständen, die erstellten Testfälle ausführen und es kann somit ein gewisses Maß an Qualität gewährleistet werden.

(Balzert 2011) (Burkhard 2012)

2.1.5 Release

Dieser Schritt bezeichnet das Installieren des finalisierten Softwaresystems in dieser Iteration, in der Produktionsumgebung des Endnutzers, hier können auch weitere Schritte durchgeführt werden, wie das Durchführen von Abnahme- und Akzeptanztests, welcher von den Endnutzer durchgeführt werden kann, oder von dem Ersteller des Softwaresystems. Im weiteren Leben der Software ist dies auch regelmäßig durchzuführen, da aufgrund des Iterativen Ansatzes immer wieder neue Softwarereleases wird.

(Balzert 2011) (Burkhard 2012)

2.1.6 Wartung

Die Installierte und in diesem Schritt letzte Version der Software muss hier gewartet werden, denn es kann leider immer wieder vorkommen das Fehler in der Testphase nicht entdeckt wurden oder Ressourcen zu knapp kalkuliert wurden, auch werden hier bereits neue Anforderungen gesammelt für die nächste Iteration der Software.

(Balzert 2011) (Burkhard 2012)

2.2 Agiles Manifest

Wie bereits in mehrere Kapitel erwähnt, ist die Grundlage für viele Agile Vorgehensmethoden das *„Agile Manifest,“* welches an einem Februar Wochenende 2001, in einer Ski Hütte in Wasatch Utah, verfasst wurde. Die Notwendigkeit für dieses Manifest sahen die Autoren da die Softwareentwicklung immer komplexer wurde, und auch der Fokus der Arbeit immer mehr auf deren Dokumentation gelegt wurde. Somit erschufen sie das Agile Manifest, dieses wurde von 17 Entwicklern entworfen, welche aus unterschiedlichen Bereichen kamen und auch vorab schon unterschiedliche Methodiken für die Softwareentwicklung erarbeiteten, zum Beispiel Schwab und Kent mit dem *„Scrum“* Guide. Hierbei sollte das Agile Manifest allerdings lediglich die grundlegenden Ideen nahelegen, welche für das Agile Arbeiten notwendig sind, anders als wie zum Beispiel Scrum welches ein Framework ist. (Kent, et al. 2001) Dafür haben die Autoren *„Agile Werte und Prinzipien“* entwickelt welche als Leitfäden dienen sollen.
Kurz zusammengefasst lässt sich aber sagen, dass die Personen, welche die Arbeit verrichten im Mittelpunkt stehen, und wie diese miteinander kommunizieren, so sollen diese auch *„Cross-Funktional“* arbeiten, es sollen also nicht Rollen erfüllt werden in einem Team, sondern Fähigkeiten („Skills“) abgedeckt werden.

2.2.1 Agilen Werte

Individuen und Interaktion mehr als Prozesse und Werkzeuge
Funktionierende Software mehr als umfassende Dokumentation
Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
Reagieren auf Veränderung mehr als das Befolgen eines Plans

(Kent, et al. 2001)

Dies sind die Vier Werte, die im Agilen Manifest niedergeschrieben wurden, mit diesen Werten versuchen die Autoren des Agilen Manifest die Probleme der Klassischen Softwareentwicklung zu beheben, oft werden diese Werte allerdings in verschiedenen Agilen Methodiken auch unterschiedlich umgesetzt. Auch sind diese Werte nicht in Stein gemeißelt und sind auch nicht als feste Werte zu Betrachten oder als strikte Regeln, sondern diese sollen eher als Leitfaden gesehen werden, mit diesem Leitfaden soll es dann möglich sein eine bessere Arbeitsweise zu schaffen in denen Software entwickelt werden kann.

Aber was bedeuten sie im Einzelnen und was ist die Idee hinter diesen? Auf diese Werte möchte ich hier nur eingehen.

2.2.1.1 Funktionierende Software mehr als umfassende Dokumentation

Früher wurde die entwickelte Software erst ausgeliefert, wenn die zugehörige Dokumentation auch fertig geschrieben wurde, welches abhängig von der Größe der Software eine beträchtliche Zeit in Anspruch nehmen konnte. Aus diesem Grund entstand dieser Agile Wert, Dokumentation soll hier nicht vernachlässigt werden, auch soll der Agile Wert hier nicht als Ausrede verwendet werden die notwendige Dokumentation nicht zu schreiben. Vielmehr soll hier die Dokumentation anhand von selbsterklärenden User-Stories geschehen, so bleibt dem Entwickler mehr Zeit für das Entwickeln der Software.

(Team 2022) (Eby 2019)

2.2.1.2 Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung

Bei der Verwendung des Wasserfall Modells, verhandelt der Kunde vor der Entstehung der Software, sowie abschließend mit dem Produktmanager, allerding nicht während dem eigentlichen Entwicklungsprozess, hier geht das Agile Manifest beziehungsweise der Agile Ansatz einen anderen weg. Hier wird dem Kunden nicht am Ende des Entwicklungsprozesses eine fertige Software präsentiert, sondern vielmehr sollen regelmäßige Gespräche stattfinden, und auch das

Einholen von Feedback so zum Beispiel durch das Veranstalten von Demonstrationen, dies kann in regelmäßigen Abständen stattfinden oder nach Bedarf.

(Team 2022) (Eby 2019)

2.2.1.3 Individuen und Interaktion mehr als Prozesse und Werkzeuge

Hier geht es darum das Menschen im Vordergrund stehen, anstatt der Verwendeten Tools, und die dahinterstehenden Prozesse, hier geht es stark um die Kommunikation zwischen Kunden und Team, so soll zum Beispiel nicht eine Komplexe Kommunikationskette eingehalten werden, wenn ein Thema mit einer E-Mail direkt erledigt werden kann. Die Kommunikation soll also organisch geschehen und bei Bedarf.

(Team 2022) (Eby 2019)

2.2.1.4 Reagieren auf Veränderung mehr als das Befolgen eines Plans

Durch den Agilen Ansatz werden Iterationen kurzgehalten, sodass Prioritäten schnell neu geordnet werden können und so auf Änderungen im Kundenwunsch oder Plan schnell umgesetzt werden können. In der Klassischen Methodik hingegen, wurden Änderungen argwöhnisch betrachtet, da diese oft mit einem großen Aufwand verbunden waren, da zum Beispiel oft eine große Anzahl an verschiedenen Abhängigkeiten bestand zu anderen Teams.

(Team 2022) (Eby 2019)

2.2.2 Agilen Prinzipien

Die 12 Agilen Prinzipien, sind Leitlinien, mit denen das *„Agile Movement“ beschrieben* wird, bei diesen Prinzipien sollen die Grundsätze des *„Agile Movement“* mit den Notwendigen Geschäftsprozessen in Einklang gebracht werden.

Auf alle aufgezählten Prinzipien im Agilen Manifest soll allerdings hier nicht eingegangen werden, da dieses zu ausführend wäre und für das Ergebnis und Verständnis dieser Arbeit nicht relevant ist. Auch werden allgemein die Agilen Werte oft häufiger behandelt als die Agilen Prinzipien.

Es sei lediglich kurz gesagt, dass die Grundprinzipien dieser Agilen Werte leicht verständlich sind, und als Rahmen dienen sollen für einen neuen Weg der Softwareentwicklung, dieser neue Weg soll in die Richtung von schnellen kurzen Prozessen gehen. Dadurch sind diese Prozesse äußerst

flexibel und können so optimal auf die Wünsche des Kunden adaptiert werden, auch ist das regelmäßige Ausliefern von kleinen Softwarepaketen an den Kunden ein wichtiger Punkt. Denn umso eher Software ausgeliefert wird, und dieses auch mit einer Regelmäßigkeit geschieht, umso höher ist die Zufriedenheit des Kunden, da er nicht lange auf eine funktionierende Software warten muss., und so ist es diesem auch möglich bei Diskrepanzen schnell einzugreifen. Auch liegt hier ein starker Fokus auf diesen „*Self Managing Teams“*, diese werten in einem späteren Kapitel behandelt (siehe Kapitel „*Grundlagen Agiles Arbeiten im Softwareprojekt*“)

2.3 Vorgehensmodelle in der Softwareentwicklung

Wie bereits im Kapitel der Ausgangslage beschrieben, nennt sich der Vorgang der Entwicklung von Software, Softwareentwicklungsprozess, gerade in größeren Projekten, wird hierfür ein Systematisches Vorgehen gewählt. Da sonst der Entstehungsprozess von Software äußerst unkoordiniert ist, welches die Wahrscheinlichkeit der Fehlerquellen massiv erhöht. Hierzu haben sich eine Reihe von Methoden entwickelt, welche einen standardisierten, Organisierten Rahmen bieten sollen, welche für die Idealen Ablauf der Entwicklung Notwendig ist, auf die wichtigsten möchte ich hier nun kurz eingehen.

Nun möchte hier auf die gängigsten beziehungsweise meist verwandtesten (siehe Abbildung 8), Vorgehensmodelle der „Klassischen“ und auch auf die Agilen Vorgehensmodelle eingehen, da hier schon die ersten großen Unterschiede erkenntlich sind, auf die Hybriden Vorgehensmodelle wird nicht eingegangen da diese Vorgehensmodelle für diese Arbeit keine Relevanz haben. Auch werden nicht alle Formen der Agilen und klassischen Vorgehensmethodik aufgezählt und besprochen, da dies ebenfalls den Rahmen dieser Arbeit sprengen würde, da es auch Mischformen von unterschiedlichen Methoden gibt wie zum Beispiel Scrum & Kanban.

2.3.1 Wasserfall Modell

Dr. Royce Winston publizierte im August 1970 in einer Zeitschrift, in der er das erste Vorgehensmodell für die Softwareentwicklung präsentierte, er benannte dieses von ihm entwickelte, und vorgestellte Modell allerdings nicht. In seinen Augen war es aber notwendig sein selbst entwickeltes Vorgehensmodell beziehungsweise seine Methodik mit anderen Personen zu teilen. Dr. Royce entwickelte Softwaresysteme, welche in der Raumfahrt verwendet wurden, da diese im Grad des Erfolges, sowie deren Funktionaler stand zum Fälligkeitsdatum, aber stark variierten, versuchte er diesem Einhalt zu gebieten, zu diesem Zweck entwickelte er eine Methodik. In dieser, von ihm entwickelten Methodik, sollen Arbeitsschritte, welche in der Softwareentwicklung durchgeführt werden, granular aufgeteilt werden umso kleinere Kompaktere Arbeitspakete zu erhalten. Dieser Prozess des Schaffens der von ihm geplanten Methodik, durchwanderte mehrere

Iterationen, bis das fertige Wasserfall Modell entstand. Dies ist wohl einer der ersten verwendeten Vorgehensmodelle in der Softwareentwicklung, hierbei werden einzelne Schritte klar getrennt voneinander und sequenziell durchgeführt, hierbei wird der nächste Schritt erst ausgeführt, wenn der voran gegangene Schritt beendet ist, somit muss ein Arbeitsschritt komplett durchgeführt werden bis das nächste Starten kann, es ist also nicht möglich während eines Schrittes, der gerade ausgeführt wird, den Vorherigen Schritt auszuführen. (Royce 1970)

Die Vorgehensweise und der Ablauf des Wasserfall Modells, ist in der unten Angeführten Abbildung („*Abbildung 3 Einfaches Wasserfall Modell* (Royce 1970)*“*) gezeigt, hier sind alle Schritte aufgelistet die laut Dr. Royce nötig waren, um am Ende des durchlaufes des Wasserfalles eine funktionierende Software zu erhalten. Diese durchzuführenden Schritte werden sequenziell durchlaufen, dies ist auch einer der größten Schwächen dieser Version des Wasserfall Modells, wird in einer der gerade durchgeführten Schritte ein Fehler entdeckt, oder ist etwa eine Änderung Notwendig, da zum Beispiel eine andere Funktion des Softwaresystems gewünscht wird, gibt es in diesem Modell keine Möglichkeit den voran gestellten Schritt zu durchlaufen. Dies ist vor allem Problematisch da erst sehr spät in der Entwicklungsphase die erstellte Software getestet wird, wird in dieser Phase also ein Fehler gefunden ist dies oft mit großen Kosten verbunden, sowohl in den Bereichen der Arbeitszeit und Last sowie mit Finanziellen Kosten, die sogenannte „*Technical Debt“.*

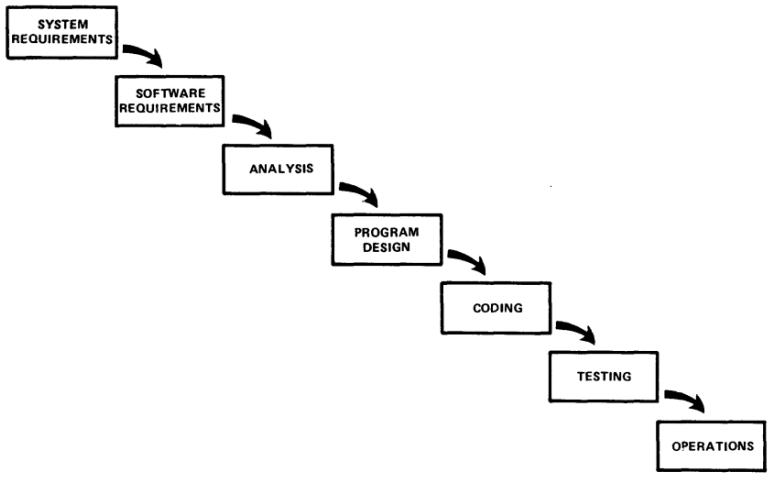


Abbildung 3 Einfaches Wasserfall Modell (Royce 1970)

Diese Schwäche der von ihm entwickelten Methodik, versuchte er mit dieser Iteration des Wasserfallmodells zu beheben, hier ist es möglich in der Implementierungskette zurückzugehen, dies ist allerdings auch nur möglich, wenn der Aktuelle Schritt abgeschlossen ist.

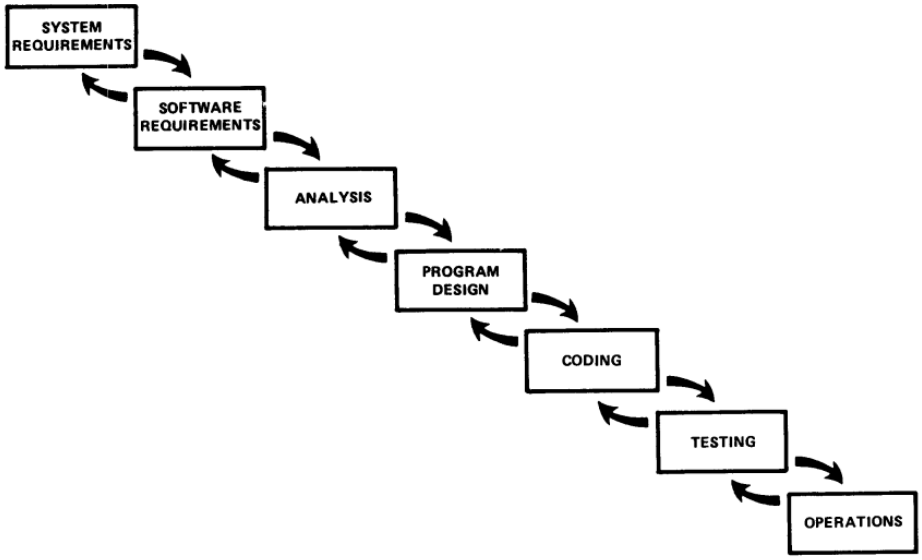


Abbildung 4 Wasserfallmodell (Royce 1970)

Trotz dieser Nachteile ist das Wasserfallmodell, wenn auf Abbildung 8 Softwareentwicklungsmethoden Verwendung (Statista 2022) geblickt wird zeigt sich das die Wasserfallmethode noch immer verwendet wird (siehe Kapitel „*Grundlagen Agiles Arbeiten im Softwareprojekte*“, „*Abbildung 10 Verwendung Agile* (Ravikumar 2023) „).

Auf die einzelnen Schritte wird hier nicht eingegangen da diese Schritte zu einem Großteil bereits beschrieben wurden in den Kapiteln „*Spezifikationsanalyse* „– „*Wartung*“ und die Funktion dieser Schritte deckungsgleich ist wie mit den Abgebildet Schritten in den Abbildung 3 & Abbildung 4.

2.3.2 Scrum

Bevor auf die Details, und den eigentlichen Arbeitsprozess der Scrum Methodik eingegangen wird, möchte ich hier auf einige der verwendeten Begriffe eingehen, die in dem Scrum Handbook verwendet werden, und diese verwendeten Begriffe erklären. Da diese verwendeten Begriffe für das weitere Verständnis, der Scrum Methodik wichtig ist, und somit auch eher eine Spezifische Verwendung dort finden. Auch wenn einige der hier verwendeten Begriffe auch allgemeiner in der Softwareentwicklungsbranche bekannt sind.

2.3.2.1 „Definition of Ready“&„Definition of Done”

Dies ist wohl eine der Wichtigsten Definition für ein funktionierendes Arbeiten in einem SCRUM Team, da durch eine gute Definition beider Begrifflichkeiten, ein flüssiges Arbeiten gewährleistet werden kann. Leider findet sich im SCRUM Guide selbst keine klare Definition dieser Begriffe

lediglich eine grobe Umschreibung, welche beschreibt das diese Begriffe klar definiert wurden, wenn keine weiteren Fragen bestehen.

2.3.2.1.1 Definition of Ready (DoR)

Die im Backlog Priorisierten Ziele für diesen Sprint, müssen einzeln definiert werden hierzu ist es wichtig festzulegen, wann ein Ticket genug allgemein Notwendige Informationen beinhaltet, dass es von einem Entwickler des Teams umgesetzt werden kann, dies ist die „*Definition of Ready*“. Hierbei geht es um allgemeine Ziele des Tickets wie zum Beispiel „Was soll umgesetzt werden“ und „Warum“, und ähnliche Punkte allerdings gibt es hier unterschiedliche Auffassungen wann ein Ziel genug Informationen beinhaltet das es als „Ready“ zählt, die Definition von Jeff Sutherland für die Definition of Ready ist wie folgt: „*ready is when the team says: ‚Ah, we got it*““.

(ScrumEvent n.d.)

2.3.2.1.2 Definition of Done (DoD)

Auch gibt es diese selbe Definition für den Zeitpunkt beziehungsweise Fortschritt, wann die Entwicklung eines Tickets eingestellt werden kann, dies ist die „*Definition of Done“*, oft wird diese Definition von dem Unternehmen vorgegeben, und zählt als der Minimum Standard für alle Teams innerhalb des Scrums, beziehungsweise des Unternehmens.

(ScrumEvent n.d.)

2.3.2.2 Rollen in Scrum

Scrum verwendet mehrere Rollen, welche für ein Scrum Team benötigt werden, jeder Rolle sind bestimmte Aufgaben zugewiesen auf diese Rollen möchte ich hier nun kurz eingehen

2.3.2.2.1 Developer

Dies sind die Mitglieder des Scrum Teams, die zuständig sind für die Abarbeitung des Backlogs, und dem Erstellen eines nutzbaren Produktes an dem Ende eines Sprintes, hierbei unterscheidet Scrum allerdings nicht nach der Tätigkeit so zählen zum Beispiel Entwickler und Tester als Developer siehe dazu Kapitel „*Scrum Prozess*“, somit sind Developer für folgende Aufgaben verantwortlich in einem Sprint:

- Planen für den Sprint
- Qualität durch die Einhaltung einer Definition of Done
- Den erstellten Plan täglich anpassen das dieser passend zu dem Sprint Ziel ist

- Feedback und Rat unter anderen Entwicklern im Team einholen

(Sutherland and Schwaber 2020)

2.3.2.2.2 Product Owner

Dieser ist verantwortlich für die Maximierung des Wertes des Produktes, welches sich aus der Arbeit des Scrum Teams ergibt, dies kann zum Beispiel erreicht werden durch das Priorisieren des Backlogs. Weitere Aufgaben des Product Owner laut Scrum Guide sind:

- Definition und Kommunikation des Produkt-Zieles
- Backlog Einträge zu erstellen und deren Klare Kommunikation
- Regelmäßige Priorisierung des Backlogs
- Sicherstellen das diese Backlog Einträge Verstand und Sichtbar sind

(Sutherland and Schwaber 2020)

2.3.2.2.3 Scrum Master

Dieser ist verantwortlich für die Implementierung und Umsetzung von Scrum, hierbei dient der Scrum Guide als Handbuch, weiters ist der Scrum Master Zuständigkeit für die Effizienz des Scrum Teams, so versucht diese eventuellen Hindernisse oder Probleme zu beseitigen.

Weitere Aufgaben des Scrum Masters laut Scrum Guide sind:

- Coaching des Teams in Selbstmanagement und Interdisziplinärer Zusammenarbeit
- Unterstützung bei der Definition of Ready und Done
- Sicherstellen das alle notwendigen Scrum Prozesse eingehalten und durchgeführt werden sowie innerhalb der Time box liegen

(Sutherland and Schwaber 2020)

2.3.2.3 Sprint

Dies ist ein festgelegter Zeitintervall von maximal einem Monat meistens allerdings Kürzer, in dieser Zeit werden die Ziele, welche in dem Sprint Planning definiert wurden, umgesetzt, ist die Länge eines Sprintes initial definiert wird diese auch nicht geändert da diese gleichbleibende Konstante Arbeitsweise auch Sicherheit und Qualität erzeugt. Während eines Sprintes definiert der Scrum Guide allerdings folgende Punkte:

- Keine Änderungen vornehmen, die das Sprint Ziel gefährden könnten
- Die Qualität nimmt nicht ab

- Bei Bedarf Korrektur des Backlogs
- Der Scope kann geklärt werden und mit den Product Owner, neu vereinbart werden

Ein Sprint ist also ein in sich geschlossener Prozess an dessen Ende dem Kunden eine Funktionierende Software beziehungsweise Funktion präsentiert werden kann.

(Sutherland and Schwaber 2020)

Alle weiteren Scrum Tätigkeiten (die unten angeführten Punkte*)* werden ebenfalls in dieser Zeit des Sprints durchgeführt, auch werden die einzelnen Prozesse Zeitlich voneinander abhängig durchgeführt, (siehe „*Abbildung* 5 Scrum Sprints (Henderson 2023)“).

Backlog

Im Backlog befinden sich die Ziele des Teams die sie sich für diesen Sprint gestellt haben, dieser wird von dem Product Owner nach Priorität sortiert, und von den Developern abgearbeitet, auch wird der Backlog während des Sprintes regelmäßig neuorganisiert und priorisiert umso die Dringlichkeit der Neusortieren Tickets darzustellen. Der Backlog sollte das einzige Ziel des Teams während eines Sprintes sein, auch erzeugt das Definieren des Backlogs ein Kommittent des Teams und ermöglich ihnen einen Fokus auf dieses Ziel.

(Sutherland and Schwaber 2020)

Sprint Daily

Das Sprint Daily findet täglich zu einer festgelegten Zeit an einem festgelegten Ort statt, die Dauer, die ein Daily brauchen sollte, liegt bei maximal 15 Minuten, das Ziel des Daily ist es, die Kommunikation zu fördern, den Fortschritt des Sprintes zu überprüfen, und bei Bedarf den Backlog anzupassen. Auch kann das Daily für einen allgemeinen Informationsaustausch verwendet werden sollte zum Beispiel ein Teammitglied Probleme bei einer Aufgabe haben kann er diese im Daily einbringen, um so schnell Hilfe zu erhalten.

(Sutherland and Schwaber 2020)

Sprint Planning

Das Ziel des Planning ist es, die Ziele für den nächsten durchzuführenden Sprint zu planen, auch lassen sich bereits in diesem Planning initiale Fragen stellen, die vor dem Start des Sprintes geklärt werden können, zu beachten ist allerdings das am Ende des Planning ein Ziel vereinbart wurde.

(Sutherland and Schwaber 2020)

Sprint Review

Das Ziel des Reviews ist es, das an dem Ende eines Sprintes das Ergebnis kontrolliert wird, hier können auch Ergebnisse, Stakeholdern vorgestellt werden, um diese so über den Fortschritt des Projektes zu Informieren und gleichzeitig Feedback einholen.

(Sutherland and Schwaber 2020)

Sprint Retroperspective

Das Ziel der Retroperspective ist es, Prozesse, Interkationen und ähnliches zu überprüfen und für den nächsten Sprint anzupassen, so ist es möglich in regelmäßigen Abständen Potenzielle Hindernisse zu beseitigen, um so die Produktivität für den nächsten Sprint zu steigern. Die Retroperspective kann allerdings auch genutzt werden, um Lob auszusprechen an bestimmten Individuen oder anderen.

(Sutherland and Schwaber 2020)

2.3.2.4 Scrum Prozess

Scrum wurde von zwei Mitbegründern beziehungsweise Autoren des „*Agile Manifesto*“ erschaffen, mit der Entwicklung der Scrum Methodik begannen die beiden im Jahre 1990, die erste Version wurde 2010 publiziert und wird seitdem weiterentwickelt, Scrum ist einer der meistverwendeten Agilen Methodiken siehe dazu „*Abbildung 8 Softwareentwicklungsmethoden Verwendung (Statista 2022)*“. Die Idee von Scrum ist es die Verantwortung an kleine Teams zuzuweisen, welche anhand ihrer Erfahrung und ihres gesammelten Wissens, die bestmöglichen Entscheidungen treffen zu können, hierbei werden die oben genannten Prozesse verwendet. Dank des sich wiederholenden Prozesses, aus kleinen Sprints, ist es den Teams so möglich, Output während eines Sprintes zu maximieren.

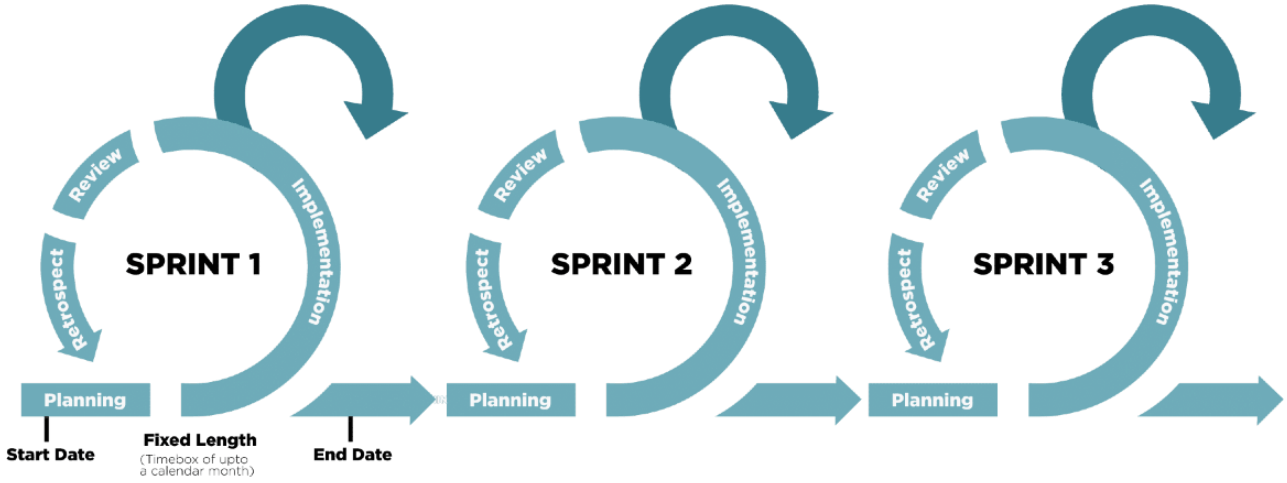


Abbildung 5 Scrum Sprints (Henderson 2023)

Wird oben angeführte Abbildung („*Abbildung 5 Scrum Sprints* (Henderson 2023)“) betrachtet, ist das Vorgehen von Scrum klar erkennbar, ein Sprint hat ein vorher definiertes Start- und Enddatum und auch somit eine vordefinierte Länge, auch findet während der Implementierung des Backlogs täglich das Sprint Daily statt. Am Ende des Sprints werden das Sprint Reviews sowie die Retroperspektive durchgeführt, und abschließend mit dem aktuellen Sprint, findet das Planning des nächsten Sprintes statt. Diese kleinen Iterationen erlauben es flexibler auf Änderungen zu reagieren, und ebenfalls den Kunden beziehungsweise den Stakeholdern zu vordefinierten Terminen ein Ergebnis zu liefern. Durch diesen engeren und Intensiveren Austausch mit den Stakeholdern findet so auch ein Austausch an Feedback statt. Anders als beim Wasserfall Model ist der Kunde in diesen Prozess aktiver involviert, und kann schneller auf Änderungen Reagieren.

(Schwaber and Sutherland n.d.) (Sutherland and Schwaber 2020) (Henderson 2023)

2.3.3 Kanban

Das Wort Kanban setzt sich aus den japanischen Wörtern „Kan“ was Visuell und „Ban“ Schild bedeutet zusammen, dies waren Schilder die Ladenbesitzer vor ihr Geschäft stellten, um so Waren und oder Dienstleistungen anzubieten, diese Kanban wurden aufwendig künstlerisch gestaltet umso mehr Potenzielle Kunden anzulocken. Dies wäre heute gleich zusetzten mit Werbeschildern oder Plakaten, auch ist der Grund Gedanke sowie der Verwendungszweck von den Kanban vergangener Zeiten sowie den Kanban der Moderne derselbe, es geht hierbei darum Inhalte klar und präzise zu kommunizieren. Kanban wurde im Jahr 1904 in Japan von entwickelt, es wurde inspiriert von der Produktion der Firma Toyota, da diese den „*schlanken Produktion*“ (lean manufacturing) Ansatz verwendeten. Sowie eine „*Just in Time*“ Produktion, dieser Ansatz wurde

von „Taiichi Ohno“ entwickelt, um Toyota aus einer Finanzielle Notlage zu helfen. In Kanban geht es sehr stark um Prozesse und deren Optimierung um so wenig Leerlauf wie möglich zu erzeugen, sowie dem Vermeiden von nicht relevanten Schritten, oft wird in der Softwareentwicklung Scrum in Kombination mit Kanban verwendet, da Scrum den allgemeinen Prozess beziehungsweise Ablauf der Entwicklung regelt, und mit Kanban oft der „*Backlog*“ visuell dargestellt wird sowie zugewiesen. (wikious n.d.) (Corona and Eros Pani 2023)

Die Verwendung von Kanban selber ist in „*Abbildung 6 Kanban Beispiel Board (eigene Erstellung)*“ ersichtlich, hierbei werden einzelne Prozesse in Abschnitte aufgeteilt, wie in diesem Beispiel zum Beispiel „*Working“.* Die einzelnen Aufgaben werden dann von Prozessabschnitt zu Prozessabschnitt geschoben, abhängig von dem Prozess, in welchem sich die Aufgaben gerade befinden. Auch erhält jede Aufgabe eine Zuständige Person, ebenfalls kann hier mit Farbcodierung gearbeitet werden, so wird schnell ersichtlich, wie die Aktuelle Arbeit last des Teams ist und eine Optimierung beziehungsweise Umverteilung von Aufgaben Notwendig ist.

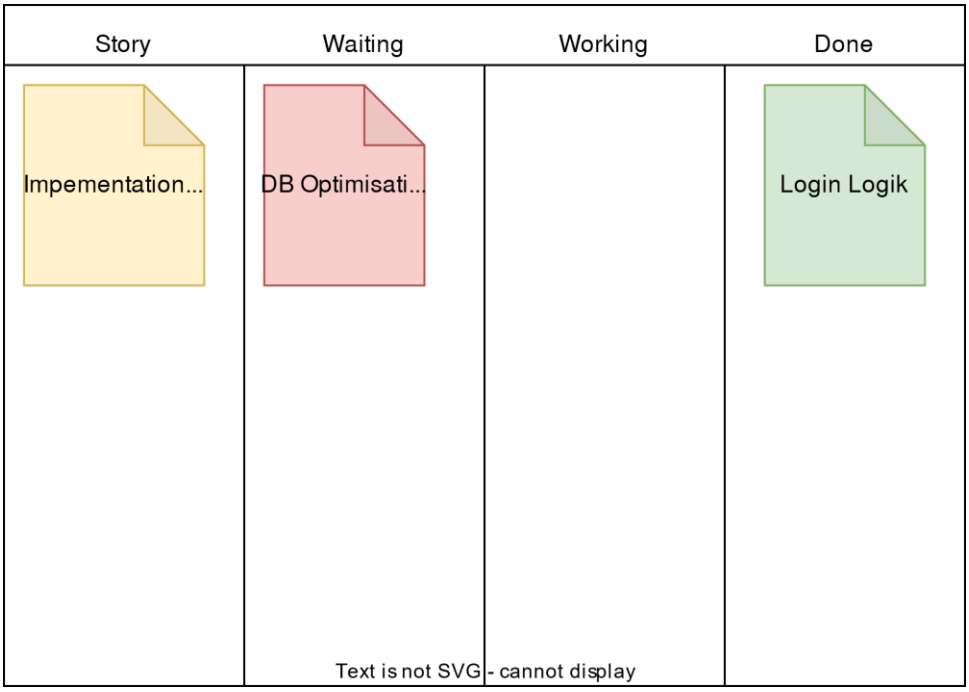


Abbildung 6 Kanban Beispiel Board (eigene Erstellung)

Somit ist die Verwendung von Kanban verglichen mit anderen Agilen Methodiken, sehr vereinfacht, aus diesem Grund wird Kanban oft verwendet auch in Kombination mit anderen Agilen Methodiken. Für die tatsächliche Verwendung eines Kanban Boards gibt es viele Möglichkeiten, so kann eine Pinnwand verwendet werden, welche mit Zetteln bestückt ist, natürlich gibt es auch hierfür diverse Softwarelösungen (zum Beispiel Jira), diese Einfachheit, und gleichzeitige

Effektivität, von Kanban sind nur wenige der Gründe warum es in der Agilen Welt so häufig verwendet wird (siehe „*Abbildung 8 Softwareentwicklungsmethoden Verwendung (Statista 2022)*“)

2.4 Potenzielle Stör- und Erfolgsfaktoren

In diesem Kapitel soll auf Potenzielle Stör- sowie Erfolgsfaktoren in der Agilen Umgebung, eingegangen werden. Es sollte klar sein das hierbei nicht alle Gründe unabhängig, ob Positiv oder Negativ betrachtet und aufgezählt werden können, da dieses zu einen nie enden wollenden Prozess führen würde. Natürlich beeinflussen sich diese zwei Aspekte auch gegenseitig, da Erfolg und Störung, von Arbeit oft nahe beieinander liegen und natürlich auch subjektiv wahrgenommen werden können, diese Wahrnehmung lässt das fällen einer allgemeinen Meinung beziehungsweise Aussage deutlich schwerer. Allerdings lassen sich einige Kernelemente immer wieder in verschiedenen Studien, ausfindig machen, und gerade das Identifizieren dieser Kernelemente ist für den Erfolg oder Misserfolg eines Projektes wichtig. Aufgrund dieser Schwierigkeiten soll hier lediglich ein überblick geschaffen werden, welche Gründe aktuell in der Literatur und Wissenschaft genannt werden, und diese kurz analysieren, da diese Literatur natürlich auch einen Einfluss in der von mir verwendeten und erstellen Umfrage gefunden hat. Anschließend wir das Ergebnis der Umfrage mit der gängigen Meinung der Wissenschaft verglichen, um festzustellen ob hier ein Konsens vorliegt oder sich in einigen dieser Punkte doch eine Diskrepanz aufweist.

2.4.1 Sektoren

Oft werden in der Literatur diese unterschiedlichen Faktoren in Kategorien unterteil um diese anschließend besser zuordnen zu können, hier gibt es eine Unterscheidung in der Anzahl sowie auch in der Bezeichnung dieser Kategorien, somit ist auch hier vieles eine Persönliche auslegen, wie bereits in der Ausgangssituation beschrieben ist diese Wahrnehmung dieser Faktoren stark von der beobachtenden Person abhängig. Allerdings findet sich in der Literatur öfter eine solche oder ähnliche Abbildung („*Abbildung 7 Faktoren (eigene Erstellung)*“) wieder, die Anzahl der Sektoren beziehungsweise Kategorien können Variieren, so zum Beispiel zählen einige Studien vier Sektoren auf (Altuwaijri and Ferrario 2022) einige andere Studien erwähnen deutlich mehrere Sektoren (Gorans and Kruchten 2014), für diese Arbeit beschränke ich mich allerdings auf die vier unten angeführten Sektoren. Auch wenn sich die Anzahl der Sektoren unterscheiden so sind die genannten Kernaspekte die gleichen, hierbei geht es lediglich um die Aufteilung dieser unterschiedlichen Sektoren, es ist hier lediglich eine zusätzlicher Detailgrad mit involviert da viele Faktoren auch unterschiedlichen Sektoren zugewiesen werden können, und die Anzahl der Sektoren auch beliebig erweitert werden.

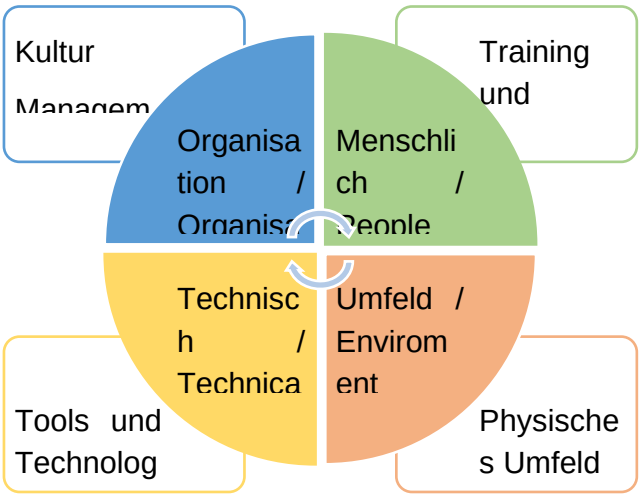


Abbildung 7 Faktoren (eigene Erstellung)

Diese Unterteilung in verschiedenen Sektoren ermöglicht es so leichter, dass Faktoren kategorisiert werden können, diese Kategorisierung ermöglicht dann das ausfindig machen von Kernsektoren welches Agiles Arbeiten wesentlich deutlicher beeinflussen als andere Sektoren. Natürlich hätten noch weitere Sektoren verwendet werden können wie der „*Process Factor*“ beschrieben in (Chow and Cao 2007), dies geschah allerdings nicht da Prozesse immer ein Teil einer Organisation sind, es gibt allerdings auch eine Argumentation das umso mehr Sektoren es gibt, umso genauer kann die Identifikation dieses Kernsektors geschehen. Allerdings könnte dann auch das gegen Argument gebracht werden, das jeder Faktor Ansicht ein einzelner Sektor sein kann und diese Aufgliederung könnte beliebig fortgesetzt werden, aus diesem Grund wurde sich auf die obigen Vier Kernsektoren begrenzt, welche in einem späteren Teil dieser Arbeit auch untersucht werden.

2.4.2 Störfaktoren

Wie bereits in dem Kapitel *“Sektoren“* findet sich hier auch das dieselbe Problematik das es in der Literatur eine Vielzahl von Unterschiedlichen genannten Störfaktoren gibt, allerdings ist die Definition dieselbe, es ist ein Faktor, der das Arbeiten in einer oder anderen Art behindert. Auch gehen diese Faktoren oft einher mit Erfolgsfaktoren, so benennen Chow und Cao einige dieser Störfaktoren (Chow and Cao 2007) die von Chow und Cao genannten Faktoren sind in der unten stehenden Tabelle („*Tabelle 1 Störfaktoren (Chow and Cao 2007)*„) aufgelistet, diese wurden von Englischen in das Deutsche übersetzt.

Sektor	Störfaktor

Organisation / organisational	● Mangelndes Sponsoring durch Führungskräfte ● Mangelnder Kommittent der Führungskräfte ● Mangelnde Agile Logistik ● Zu Traditionelle Organisationskultur ● Zu Politische Organisationskultur ● Zu große Organisation
Menschlich / People	● Mangelnde Fähigkeiten ● Mangelnde Projekt Management Kompetenzen ● Mangelnde Teamfähigkeiten
Prozess / Process	● Schlecht definierte Projekt Scope ● Schlecht definierte Projekt Anforderungen ● Schlecht definierte Projekt Planung ● Schlecht definierte Kunden Rollen ● Mangelnde Agile Fortschritts Dokumentation ● Mangelnde Kunden Kommunikation
Technisch / Technical	● Mangelnde Anwendung von Agilen Methoden ● Falsche Nutzung von Technologie und Werkzeugen

Abbildung 8 Störfaktoren (Chow and Cao 2007)

Bei der Betrachtung der Tabelle ist schon ersichtlich das Chow und Cao für jeden Sektor (auch wenn diese andere Sektoren Benennungen haben wie im weiteren Verlauf der Arbeit) eine Unterschiedliche Anzahl an Faktoren benennen. Auffallend ist das der Technische Aspekt laut Chow und Cao lediglich zwei Faktoren aufweist, die Sektoren „*Organisation & Prozess“* aber die größte Anzahl an Faktoren aufweist, anhand der Anzahl der Gelisteten Faktoren lässt sich schon ein Schluss schließen welche Sektoren hier den größten Einfluss nehmen. (Chow and Cao 2007) Diese angeführte liste könnte beliebig erweitert werden da eine große Anzahl an Störfaktoren gefunden werden könnte.

2.4.3 Erfolgsfaktoren

Ebenso ein wichtiger Punkt für den Erfolg des Agilen Arbeitens wie die Störfaktoren sind die Erfolgsfaktoren, auch hier finden sich in der Literatur eine Vielzahl von genannten Faktoren. So zum Beispiel betrachtet eine Studie aus 2016, (Yngve, et al. 2016) den Zusammenhang zwischen Teamwork Qualität und den Erfolg von Agilen Teams, diese kam zu dem Ergebnis, das eine gute Qualität der Teamarbeit zu einer höheren Produktivität des Teams führt, welches weiter zu dem Erfolg des Teams beiträgt. Auch sind diese Faktoren von einer Vielzahl von Variablen abhängig, auch kann eine Unterschiedliche Definition von Erfolg vorliegen welches zu einer weiteren Verzerrung von Erfolg führt. In der Studie von Chow und Cao wurden 36 Erfolgsfaktoren definiert, die sich über 5 Sektoren verteilen (Chow and Cao 2007), wohingegen Fahad und Ferrario lediglich 12 Faktoren nennen, verteilt auf 4 Sektoren (Altuwaijri and Ferrario 2022). Chow und Cao nannten in ihrer Studie die drei meistgenannten Erfolgsfaktoren mit:

- Delivery Strategy
- Agile Software Engineering Practices
- Team Capability

Fahad und Ferrario nannten allerdings folgende Faktoren:

- Team Capability
- Customer Involvement
- Training and Learning

Somit ist also eine Unterschiedliche Wahrnehmung dieser Faktoren festzustellen, doch lässt sich hier ein gemeinsamer Faktor der Teamfähigkeit erkennen. Allerdings können hier nicht auf alle Potenziellen Erfolgsfaktoren eingegangen werden da auch dieses zu einer nie endenden Liste führen würde.

3. Grundlagen Agiles Arbeiten im Softwareprojekt

In dem Kapitel „*Vorgehensmodelle in der Softwareentwicklung*“, wurde bereits auf die Theorie und Geschichte der meistverwendeten Vorgehensmodelle der Klassischen und Agilen Methodik eingegangen. Nicht nur die dort aufgezeigten unterschiede und die in diesem Kapitel bereits genannten Vorteile der Agilen Softwareentwicklung machen Agile Methodiken so beliebt, die Wichtigkeit von Agilen Methoden zeigt auch die unten Angeführte Abbildung („*Abbildung 8 Softwareentwicklungsmethoden Verwendung (Statista 2022)“*), diese Abbildung wurde der Statista entnommen (Vailshery 2022). Hier zeigt sich deutlich, das Agile Methodiken, 2022 eine starke Verwendung fanden, und es ist davon auszugehen, dass dieser Trend erhalten bleibt. Lediglich die Wasserfall Methodik findet sich hier, welche als nicht Agile Methodik zählt. Auch zeigt diese Statistik das Scrum und Kanban die meistverwendeten Agilen Methodiken sind, die Gründe hierfür sind unterschiedlicher Natur, auf die Details dieser beiden Agilen Methodiken soll aber hier nun nicht noch einmal eingegangen werden, somit sind genauere Details bitte den Kapiteln „*Scrum“* und „*Kanban*“ zu entnehmen, auf die abgebildete DevOps Methodik wird nicht eingegangen.

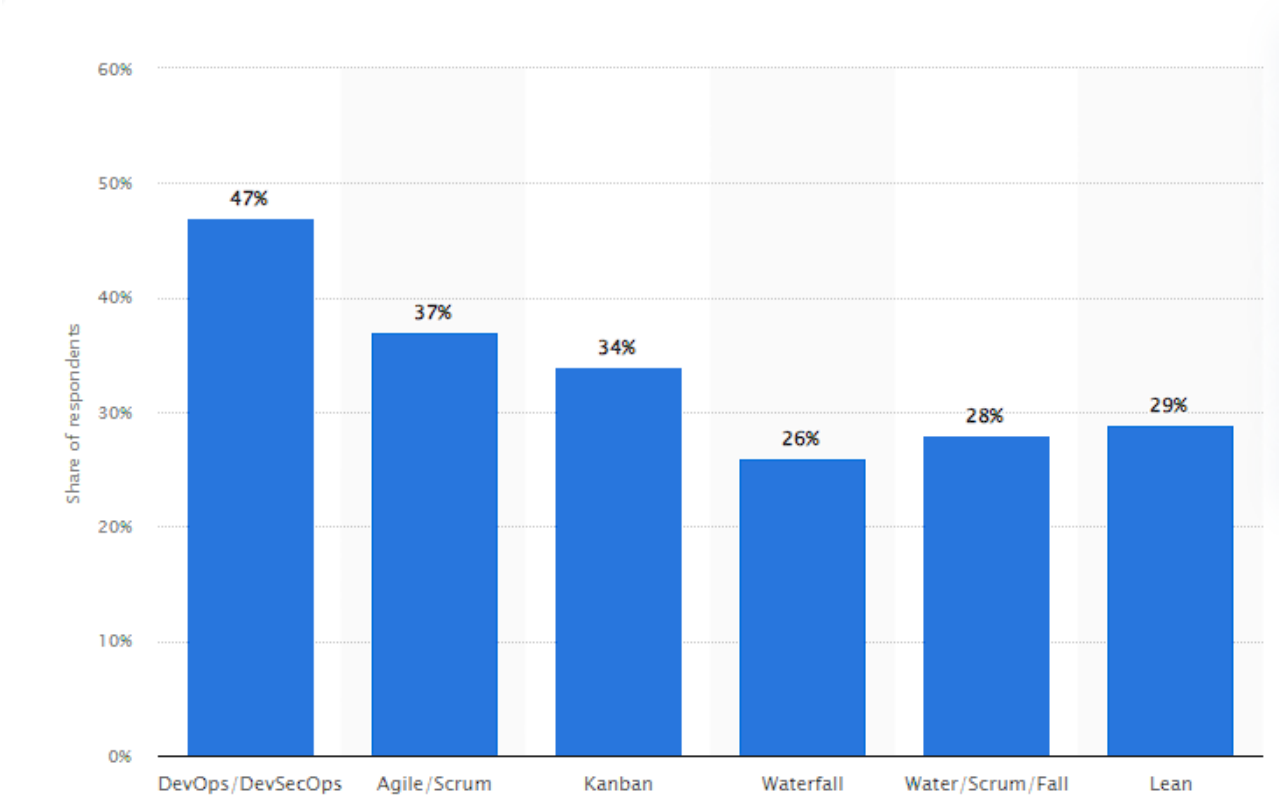


Abbildung 9 Softwareentwicklungsmethoden Verwendung (Statista 2022)

In einem Bericht von Radix (Ravikumar 2023) wird auch berichtet das 86% aller Entwickler Weltweit, die Agil Methodik, als eine „*Best-Practice*“ Methodik betrachten. Des Weiteren schreibt Ravikumar in seinem Bericht auch, dass 42% berichteten das sobald eine Agile Methodik verwendet wurden eine Verbesserung der Softwarequalität festzustellen ist, und 47% stellten fest das die Kommunikation in ihrem Team einer deutlichen Verbesserung unterzogen war.

Auch stellten diese 47% eine verbesserte Zusammenarbeit zwischen IT- und Management Team fest, welches durch die Einbringung von agilen Arbeitsweisen zurückzuführen ist. Allerdings findet der Agile Ansatz, wenn die untenstehende Grafik betrachtet wird („*Abbildung 9 Agile Verwendung außerhalb von IT* (Ravikumar 2023)“) auch immer mehr Verwendung in nicht IT, bezogenen Bereichen, wie Human Ressource und Marketing, dies zeigt einen klaren Trend dafür das Agile Methodiken im Berufsleben immer wichtiger werden, auch unabhängig von der Branche.

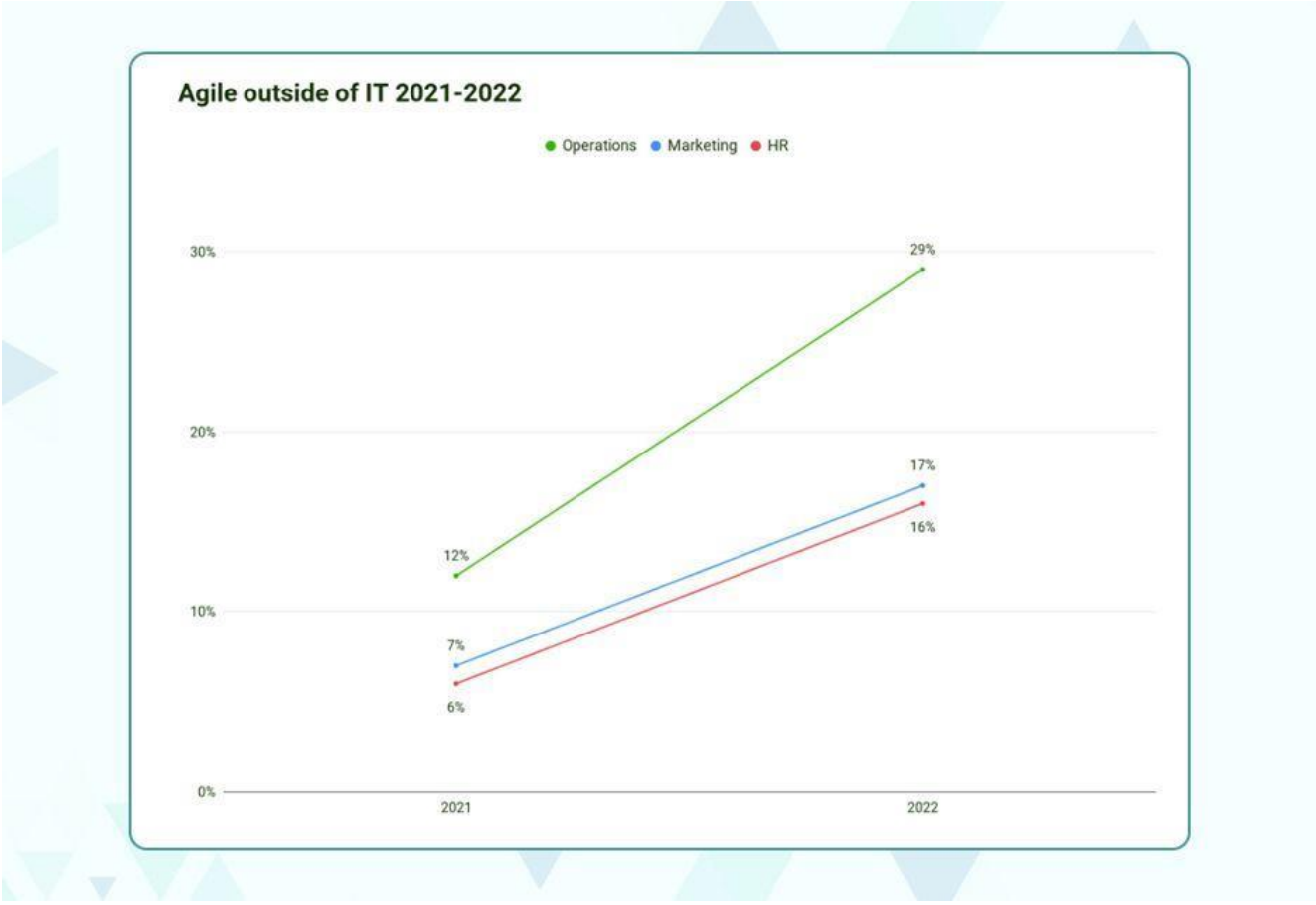


Abbildung 10 Agile Methodiken in nicht IT-Abteilungen (Ravikumar 2023)

Auch behauptet Ravikumar in seinem Bericht das (Ravikumar 2023) wenn wir nur die von ihm abgebildete Graphiken betrachten („*Abbildung 12 Agile Erfolgsrate* (Zippia 2022)

Abbildung 13 Wasserfall Erfolgsrate (Zippia 2022)“) und die dort gelisteten Werte betrachten, das lediglich 9% aller Projekte, welche eine Agile Methodik verwenden Scheitern, wobei hingegen 29% aller Projekte scheitern, welche noch die Klassische Wasserfall Methodik verwenden. Welches eine dreifach höhere Quote des Scheiterns bedeuten würde (Zippia 2022). Dies lässt den Entschluss zu das allein das verwenden, einer Agilen Methodik die Potenzielle Erfolgsquote, eines Projektes, massiv erhöhen würde, ohne dass noch weitere Erfolgsfaktoren mit in diese Quoten einbezogen wurden.

Dies würde auch die vermehrte Nutzung von Agilen Methodiken ersichtlich in „*Abbildung 10 Verwendung Agile*“ erklären, das allgemeine Interesse an der Agilen Softwareentwicklung spiegelt sich auch in der Anzahl der erschienen wissenschaftlichen Publikationen über Agilität wider (siehe „*Abbildung 11 Anzahl der Publikationen*“).

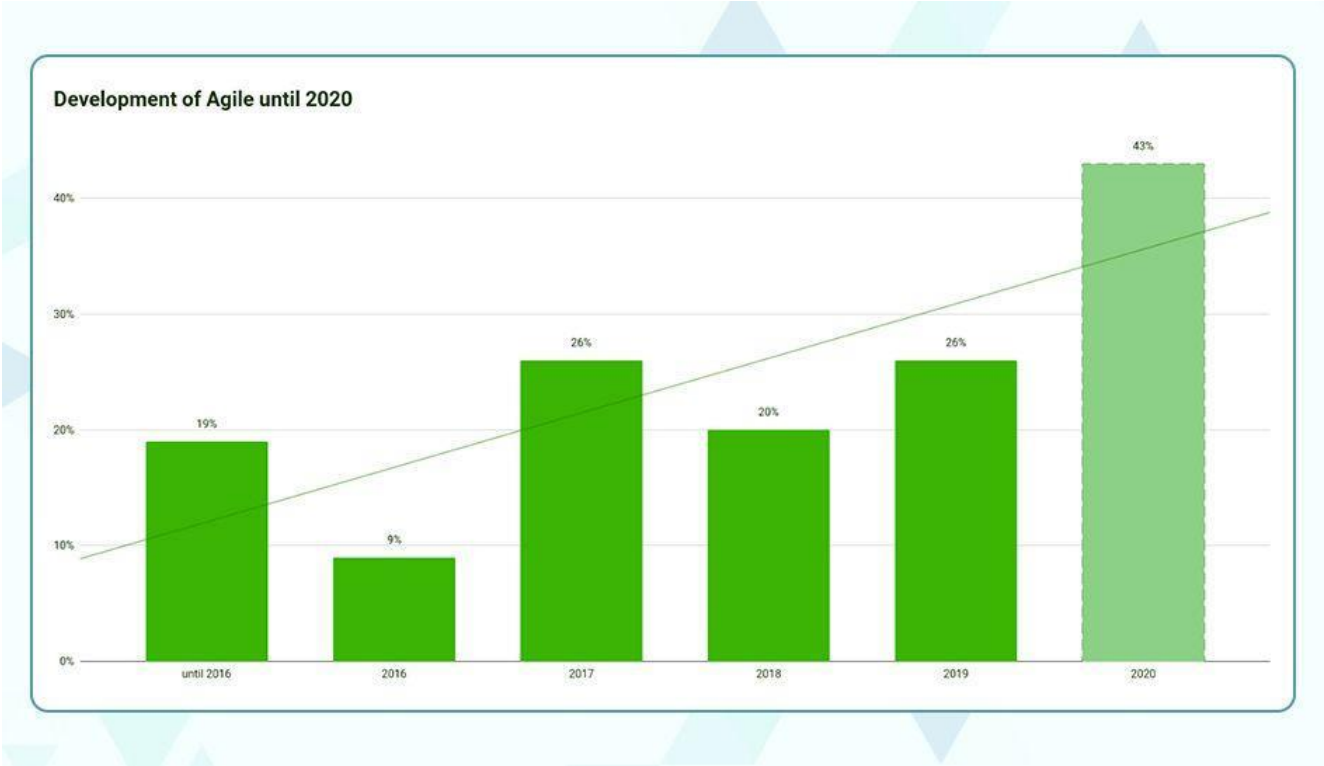


Abbildung 11 Verwendung Agile Methodik (Ravikumar 2023)

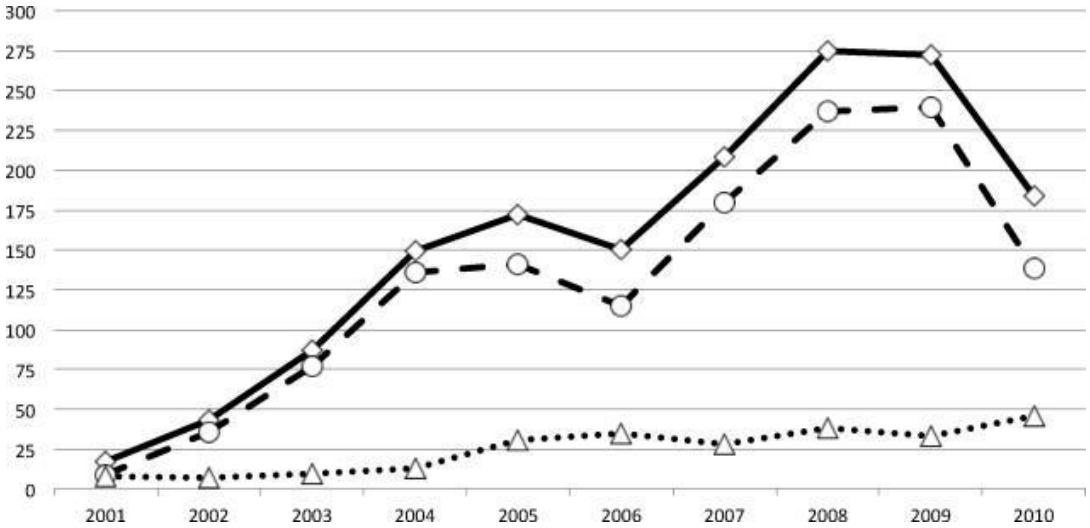


Abbildung 12 Anzahl der Publikationen (Torgeir, et al. 2012)

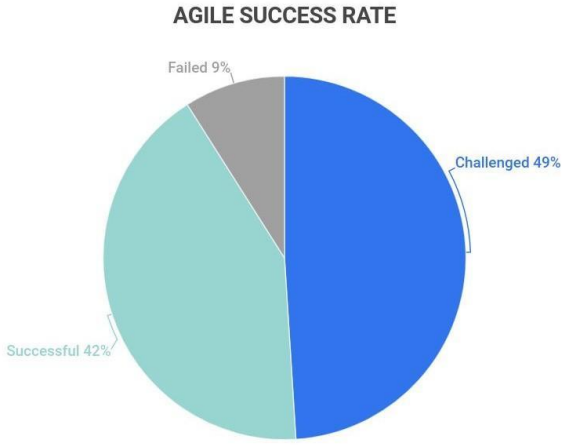
Gesamtanzahl an Publikation

Anzahl Conference Papers

Journal Articles

Dies wird noch einmal verdeutlicht angezeigt an den unten stehen Abbildungen („*Abbildung 12 Agile Erfolgsrate* (Zippia 2022)

Abbildung 13 Wasserfall Erfolgsrate (Zippia 2022*)*, leider sind die Rohdaten für diese Statistik nicht einsehbar, auch wurde keine Definition gegeben wann ein Projekt als Gescheitert (Failed) definiert wurde, nichts destotrotz sind diese Zahlen aussagekräftig.



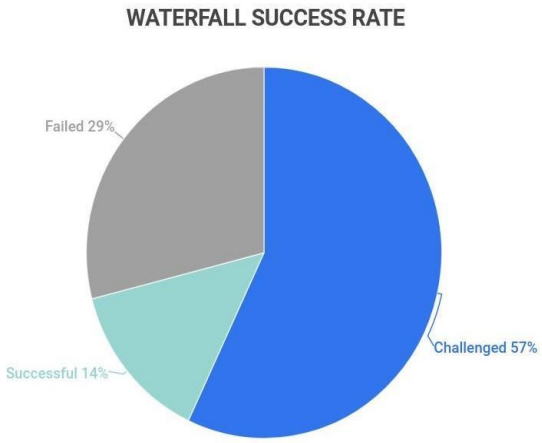


Abbildung 13 Erfolgsrate Wasserfall Methodik (Zippia 2022)

Abbildung 14 Erfolgsrate Agile Methodik (Zippia 2022)

Nur aufgrund der bis jetzt genannten Zahlen, wirkt es so, dass das Verwenden der Agilen Methodik die Wahrscheinlichkeit das ein Projekt erfolgreich abgeschlossen wird, deutlich erhöht. Allerdings stellt sich dann die Frage, wieso noch Unternehmen die Wasserfall Methodik verwenden, welches ersichtlich ist, wenn „*Abbildung 8 Softwareentwicklungsmethoden Verwendung (Statista 2022)*“ betrachtet wird. Primär wird in der Agilen Methodik der Fokus auf zwischenmenschliche Beziehung gelegt, sowohl intern als auch extern, auch wird hier viel Verantwortung an das Team übergeben, anstatt das ein Micromanaging des Managements übernommen wird. Dies soll den Vorteil haben das die Agilen Teams, am besten ihre Kapazitäten und Fähigkeiten abschätzen können. Ein großer Fokus des Agilen Arbeitens ist Flexibilität und Zwischenmenschliche Beziehungen, da dies so zu einer besseren Softwarequalität führt, und auch zu einer besseren Zusammenarbeit mit dem Kunden.

4. Forschungsmethode

Für die Belegung der aufgestellten Hypothese, verwende ich eine Umfrage, sowie ein Leitfaden-gestütztes Interview, die Zielgruppe für diese Befragung setzt sich zusammen, aus Personen, die im Agilen Softwareentwicklungsumfeld arbeiten beziehungsweise gearbeitet haben. Da diese Personen, aufgrund ihrer Tätigkeit im Agilen Umfeld eine zu berücksichtigende Expertise gesammelt haben. Die Methode der Befragung wurde ausgewählt da durch das Beantworten des Fragebogens, Daten erhoben werden können, und mithilfe dieser Daten ist es dann möglich die von mir gestellte Hypothese zu bestätigen beziehungsweise zu widerlegen.

Dieser von mir erstellte Fragebogen wird den Mitstudierenden, der Ferdinand Porsche FernFH , sowie meinem näheren Umfeld (Arbeit und Privat) zur Verfügung gestellt, umso eine Anzahl an möglichen Antworten zu erhöhen, welches die Qualität der Auswertung erhöhen soll. Für die gewählte Formen der Umfrage wurde die Online-Umfrage, sowie die Persönliche-Umfrage gewählt, die Begründung wieso sich für diese Umfragearten entschieden wurden, sind erläutert in einem späteren Teil dieses Kapitels. Allerdings handelt es sich bei der erstellten Online-Umfrage, sowie bei den Leitfaden-gestützten Interviews um standardisierte Quantitative Umfragen (wie bereits oben beschrieben erfolgt die Begründung und Erläuterung in einem späteren Kapitel). (Leipzig n.d.)

Die Antwortmöglichkeiten auf die Umfrage findet auf einer Skala statt welche eine Reichweite von 1-10 aufweist, wobei 1 die wenigste Gewichtung bedeutet, und 10 im Umkehrschluss die höchste Gewichtung, diese gesammelten Antworten werden dann analysiert und mithilfe einer Häufigkeitsanalyse ausgewertet. Mithilfe der Häufigkeitsanalyse ist es dann so möglich die drei häufig genannten Erfolgsfaktoren der Befragten festzustellen, welche dann die von mir gestellte Hypothese belegt, oder widerlegt.

4.1 Fragebogen und Interview-Leitfaden

Da sich ein Großteil dieser Arbeit mit einer Umfrage befasst, sowie der Auswertung dieser Umfrage beschäftigt, möchte ich in diesem Kapitel auf die Grundlagen von Umfragen, sowie deren Auswertungen eingehen. Dieses nahelegen der Grundlagen vermeidet Verwirrung, und schafft ein Verständnis wieso die oben genannten Methoden und Auswertungen verwendet wurden.

4.1.1 Quantitative Umfrage

Hier geht es um eine große Zahl von Antworten, sowie deren Repräsentativität und Vergleichbarkeit, dies sind oft Standardisierte Umfragen, auch finden sich hier selten offene Fragen. So wird mit der Quantitativen Umfragen oft eine große Anzahl an Daten erhoben, da hier

selten ein Expertenwissen nötig ist, und es nicht um Rahmen Bedingungen geht, sondern lediglich um das Erheben von Informationen. Die Quantitative Umfrage wird zum Beispiel kurz vor Wahlen durchgeführt, um Tendenzen in der Bevölkerung zu erkennen, der „*Warum“* Faktor wieso die genannte Partei gewählt wird, ist irrelevant.

Da es in dieser Arbeit um eine Datenerhebung geht und auch um die Häufigkeit der gegebenen Antworten ist die Quantitative Umfrage eine geeignete Methode, für die Beantwortung der von mir gestellten Hypothese.

(Leipzig n.d.)

4.1.2 Mögliche andere Methode

Eine weitere Methode der Befragung wäre die „*Qualitative Umfrage“* gewesen hier ist oft eine kleine Anzahl an befragten vorhanden, die Antwortmöglichkeiten auf jede Frage sind eher offen. Oft werden diese Daten in Experteninterviews erhoben, somit geht es hier nicht um die Anzahl der Antworten, sondern um die Qualität dieser gegebenen Antworten. Allerdings auch um die Rahmenbedingungen dieser Antworten, es wird also ein breiteres Spektrum als nur die Antwort der Befragten betrachtet. Diese Form der Umfrage eignet sich öfter, wenn nicht genügend vorhandene Literatur besteht, sowie ein breiteres Themenfeld erfragt werden soll. Aufgrund dieser Relation und dem Umfang der Antworten, die gegeben werden können, ist diese Methode für diese Arbeit nicht relevant und kann somit vernachlässigt werden, und wurde somit nicht ausgewählt für die Beantwortung der Hypothese.

(Leipzig n.d.) (Eichtstätt-Ingolstadt n.d.)

4.1.3 Fragestellungen

Um Antworten zu erhalten und somit Daten erfassen zu können, müssen Fragen gestellt werden, allerdings gibt es hier einige Unterschiedliche Methoden und Arten diese zu Fragen zu Stellen und es geben diese auch andere Antworten.

4.1.3.1 Verwendete Fragenstellungen

Nun möchte ich auch auf die in der Umfrage verwendeten Fragestellungen eingehen sowie einer kurzen Erklärung, wieso diese Typen ausgewählt wurden, und andere Typen für diese Arbeit ungeeignet sind.

Geschlossene Fragen

Hier werden die Antworten eingegrenzt, ein Klassisches Beispiel ist die „*Ja“* oder *„Nein“* Frage, allerdings ist es nicht auf diese Begrenzt, eine Geschlossene Frage bezeichnet lediglich das Vorgeben der möglichen Antworten. Diese Fragestellungsart wird in meiner Umfrage verwendet, um die Rolle der befragten Person festzustellen.

(Franziska 2018)

Skalenfrage

Hier kann einer Antwort einer Gewichtung zugeteilt werden, diese Gewichtung wird auf einer Skala dargestellt, eine Klassisches Beispiel für diesen Typ sind zum Beispiel:

„Wie zufrieden waren Sie mit dem Service: 1 sehr – 10 Gar nicht“

Aufgrund dieser erteilten Gewichtung, welche für jede Frage gegeben werden kann, können hier Schwerpunkte identifiziert werden. Durch diese Schwerpunkt Identifikation, ist diese Art der Fragestellung der primär verwendete Typ in meiner Umfrage. Auch wurde sich für eine Skalenbereich von „*1-10“* entschieden, um den Befragten so eine möglichst große Auswahl zu ermöglichen, und durch den erhöhten Skalenbereich, soll auch das „*Clustern“* von Antworten verringert werden, und somit eine bessere Aussagekraft der Antworten erzielt werden.

(Franziska 2018)

Filterfragen

Hier werden Vordefinierende Fragen gestellt welche, wenn die „*Richtige*“ Antwort getroffen wurde, eine Folgefrage gestellt werden kann, denn es macht zum Beispiel keinen Sinn jemanden über den Autokauf zu befragen, wenn dieser gar keines Besitzt beziehungsweise möchte. Solche Fragen erlauben es also die Umfrage mit einer gewissen „*Logik*“ auszustatten. Dies wird in meiner Umfrage Initial verwendet, um festzustellen, ob die Befragten in einem Agilen Softwareentwicklungsumfeld arbeiten oder gearbeitet haben.

(Franziska 2018)

4.1.3.2 Mögliche Andere Fragenstellungen

Offene Fragen

Hier ist die Antwortmöglichkeit offen, beziehungsweise kann über einen Freitext auf diese Frage geantwortet werden, dies kann verwendet werden, wenn zum Beispiel eine detaillierte Antwort erwünscht ist, dies wird oft bei der Qualitativen Umfrage verwendet. Diese wird in dieser Arbeit

allerdings nicht verwendet. Dies ist für diese Arbeit ein nicht relevanter Typus da es hier um die Gewichtung der gestellten Fragen geht, auch würde diese Form der Fragestellung die Auswertung der Umfrage erschweren. Aus diesen Gründen wurde sich gegen die Offene Frage entschieden.

(Franziska 2018)

4.1.4 Auswertungen

Die erhobenen Umfragen müssen auch ausgewertet werden, da sonst der Sinn der Umfrage fraghaft ist, hierfür stehen mehrere Optionen zur Auswahl, abhängig davon um welche Art der Umfrage es sich handelt sowie welche Aussage mit dieser Auswertung getroffen werden soll, stehen mehrere Möglichkeiten zur Auswahl.

4.1.4.1 Häufigkeitsverteilung

Hier wird die Häufigkeit der gewählten Ergebnisse gezählt, aus dieser Tabelle lassen sich dann die Häufigkeiten ablesen, so zum Beispiel wie oft eine Antwort mit „*Ja*“ beantwortet wurde. Da in dieser Arbeit die Häufigkeit der gegebenen Antworten betrachtet wird, ist diese Form der Auswertung ausgewählt worden.

(Fladerer 2023)

4.1.4.2 Mögliche andere Analysen

Allerdings gibt es noch eine weitere Auswahl an Analysen, die für die Auswertung der Erhobenen Daten verwendet hätten, werden können, allerdings aus einem oder mehreren Gründen nicht ausgewählt wurden für diese Arbeit.

Korrelationsanalyse

Hier wird der grad von Zusammenhängen zwischen Variablen festgestellt, hierbei gilt es zu unterscheiden zwischen einer Positiven und einer Negativen Korrelation, bei einer Positiven Korrelation steigen beide Variablen an, bei einer Negativen Korrelation bedeutet das Zunehmen beziehungsweise der Anstieg von Variable 1 eine Abnahme beziehungsweise Abfall von Variable 2. Diese Methode ist ungeeignet da hier jeder Faktor als einzelner Aspekt betrachtet wird und nicht auf eine Korrelation zwischen diesen Faktoren.

(Benning 2019)

Kreuztabelle

Die Auswertung dieser Ergebnisse kann hier sowohl waagerecht als auch senkrecht erfolgen. Auch hier werden Beziehungen innerhalb der Daten betrachtet, welches nicht Ziel dieser Arbeit ist.

(Fladerer 2023)

Faktorenanalyse

Hier wird versucht alle verwendeten Ausgangsvariablen auf ähnliche Nenner zu bringen und anschließend diese einem Faktoren zu zuordnen. Da die Hypothese sich mit den Meisten genannten Erfolgsfaktoren beschäftigt und nicht mit dem gemeinsamen Faktor dieser Erfolgsfaktoren ist auch diese Art der Analyse nicht anwendbar.

(Fladerer 2023)

5. Durchführen der Befragung und Interviews

Generelles

Die Umfrage wurde den Mitstudierenden der „*Ferdinand Porsche FernFH*“ sowie meinem Umfeld zur Verfügung gestellt. Sie konnte online ausgefüllt und in einem persönlichen Interview beantwortet werden. Die Fragebögen waren hierbei dieselben. Dies sollte garantieren, dass unabhängig von der Plattform, auf welcher die Umfrage durchgeführt wurde, dieselben Fragen gestellt wurden. Es wurden keine personenbezogenen Daten erhoben, um eine Anonymität der Befragten zu gewährleisten. Insgesamt nahmen an der Umfrage 60 Personen teil. Die Umfrage stand 2 Wochen zur Verfügung und die Befragten hatten kein Zeitlimit zum Ausfüllen ihres Fragebogens. Fragebögen wurden nur gewertet, wenn diese vollständig ausgefüllt wurden, und wenn die Person in einem agilen Softwareentwicklungsumfeld gearbeitet hat oder zurzeit in diesem arbeitet. Bei den Antwortmöglichkeiten, die gegeben werden konnten, handelte es sich um Ordinalskalen mit einer Reichweite von 1–10, wobei 1 die wenigste Gewichtung darstellt und 10 die höchste. Lediglich bei der Erfragung der Rolle sowie darüber, ob der Befragte gerade oder in der Vergangenheit in einem agilen Umfeld gearbeitet hat, handelte es sich nicht um Skalenniveaus. Die gestellten Fragen sind in Kapitel „*9.2 Fragenkatalog*“ aufgelistet.

Onlineplattform

Die Umfrage wurde in „*Microsoft Forms*“ erstellt und durchgeführt, da „*Microsoft Forms*“ alle notwendigen Funktionen besitzt, welche für das Durchführen der Umfrage benötigt wurden. Die Erstellung und das Ausfüllen des Fragebogens war sehr intuitiv. Auch steht es kostenfrei im Office-365-Paket zur Verfügung. So stellt es eingebaute Funktionen zur Verfügung für das Auswerten der Umfrage und das Exportieren zu einer „*Microsoft Excel*“-Datei. So wurden die unten angeführten Grafiken aus „*Microsoft Forms*“ entnommen. Die Umfrage konnte sowohl von einem mobilen Endgerät als auch von einem Computer durchgeführt werden. Es wurden auch andere mögliche Tools in Erwägung gezogen, allerdings erwiesen sich diese als zu kompliziert, nicht anwenderfreundlich oder waren mit Kosten verbunden.

Interviews

Die Personen, die in einem Interview befragt wurden, erhielten denselben Fragenkatalog wie die Personen, die online an der Umfrage teilnahmen. Der Fragebogen wurde den Befragten in Papierform zur Verfügung gestellt, von diesen ausgefüllt und anschließend manuell von mir in „*Microsoft Forms*“ übertragen. Auch handelte es sich um dieselben Antwortmöglichkeiten sowie die gleiche Reihenfolge der Fragen. Dies sollte gewährleisten, dass hier keine Verfälschung der Ergebnisse erfolgt.

6. Analyse der Ergebnisse

In dem folgenden Kapitel möchte ich nun auf die Ergebnisse der Umfrage eingehen sowie auf einzelne Fragen. Aufgrund der Anzahl der Fragen kann leider nicht auf jede Frage individuell eingegangen werden. Aus diesem Grund werde ich nur auf ausgewählte Fragen eingehen. Auch handelt es sich bei dem Ergebnis dieser Umfrage lediglich um eine Stichprobe und spiegelt somit nicht die Grundgesamtheit wider.

6.1 Auswertungsmethodik

Die drei meistgenannten Faktoren wurden mithilfe des *„Mittelwertes“* ausfindig gemacht. Initial sollte dies mit dem „*Modus*“ erfolgen. Dieser stellte sich allerdings als nicht eindeutig genug heraus, da es beim Auswerten der Umfrage mehrere gleich häufig genannte Faktoren gab. Aus diesem Grund wurde der Mittelwert ausgewählt, da so ein eindeutiger Wert ermittelt werden kann.

6.2 Auswertungstool

Für die Auswertung wurde „Microsoft Excel“ verwendet, da es alle Funktionen bereitstellt, welche benötigt wurden, um diese Umfrage auszuwerten. So stellt es eine breite Sammlung an bereits existierenden Formeln, welche für die Berechnung von Kennwerten wichtig sind, zur Verfügung, sowie die Möglichkeit zum Erzeugen von Diagrammen. Auch steht es ebenfalls kostenfrei im Office-365-Paket zur Verfügung. Die verwendeten Formeln, Rohdaten sowie der verwendete Fragenkatalog werden in einem eigenen Kapitel „*9 Formeln, Fragenkatalog und Rohdaten*“ aufgelistet.

6.3 Einleitungsfragen

Wie bereits kurz beschrieben wurden nur Fragebögen gewertet von Personen, welche zurzeit oder in der Vergangenheit in der agilen Softwareentwicklung gearbeitet haben, sowie vollständig ausgefüllte Fragebögen.

Um zu überprüfen, ob eine Person in der agilen Softwareentwicklung tätig ist, wurde eine Filterfrage eingebaut, welche mit *„Ja“* oder *„Nein“* beantwortet werden konnte. Wurde diese Frage mit „*Nein*“ beantwortet, wurde die Umfrage abgebrochen, wurde mit „*Ja*“ geantwortet, ging die Umfrage weiter. Dieses Ergebnis ist in „*Abbildung 15*“ ersichtlich und zeigt klar, dass ein Großteil der Befragten nicht das Kriterium erfüllt hat, um an dieser Umfrage teilzunehmen. Somit wurden nur die Antworten der Personen gewertet, welche diese Frage mit „*Ja*“ beantwortet haben. Die Vollständigkeit der Fragebögen wurde überprüft durch „*Microsoft Forms*“ und scheint auch hier

nicht auf, es ist also leider nicht bekannt wie viele Personen die Umfrage nicht vollständig ausfüllten.

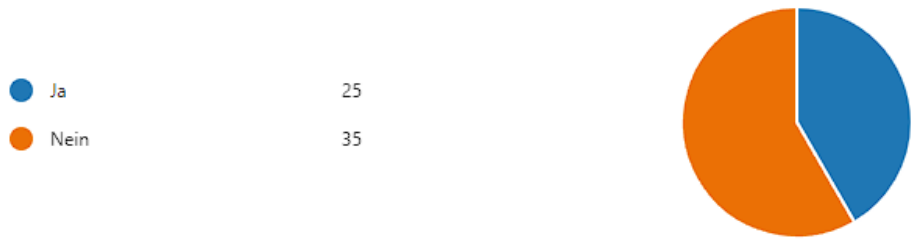


Abbildung 15 Teilnahmefrage

Auch wurde die Rolle der Teilnehmer im Team erfragt. Dieses Ergebnis zeigt sich in „*Abbildung 16* Rollenfrage“. Somit könnten Rückschlüsse zu den gegebenen Antworten und der Häufigkeitsverteilung der Rollen gezogen werden. Hier lässt sich aber eindeutig feststellen, dass ein Großteil der Befragten als „*Business Analysten*“ beschäftigt ist, gefolgt von „*Product Owner*“. Interessanterweise ist die Rolle des „*Entwicklers“* seltener vertreten als initial angenommen. Trotz dieser geringen Teilnahme an Entwicklern wurden, einigen technische Faktoren häufig genannt was, doch interessant ist.

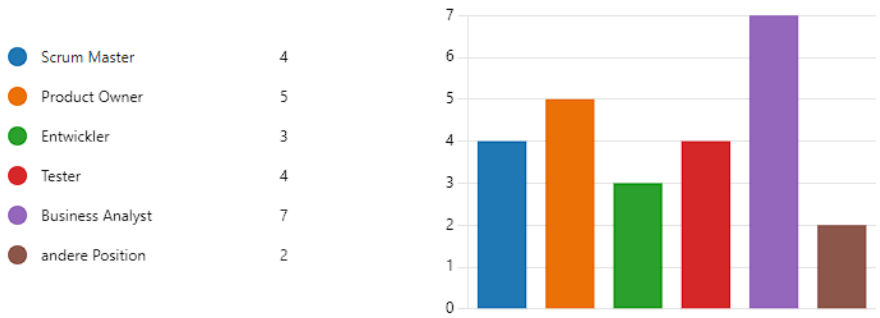


Abbildung 16 Rollenfrage

6.4 Gesamtauswertung

Für jede einzelne Frage wurde der Mittelwert errechnet, mit diesem wurde ein Diagramm erstellt. Mithilfe von „*Abbildung 17*“ lassen sich die folgenden drei meistgenannten Faktoren erkennen:

- Frage 11 mit einem Mittelwert von 9,2 (Gold)
- Frage 10 mit einem Mittelwert von 8,96 (Silber)
- Frage 21 mit einem Mittelwert von 8,84 (Bronze)

Jeder dieser drei Faktoren wird in seinem eigenen Kapitel gesondert behandelt.

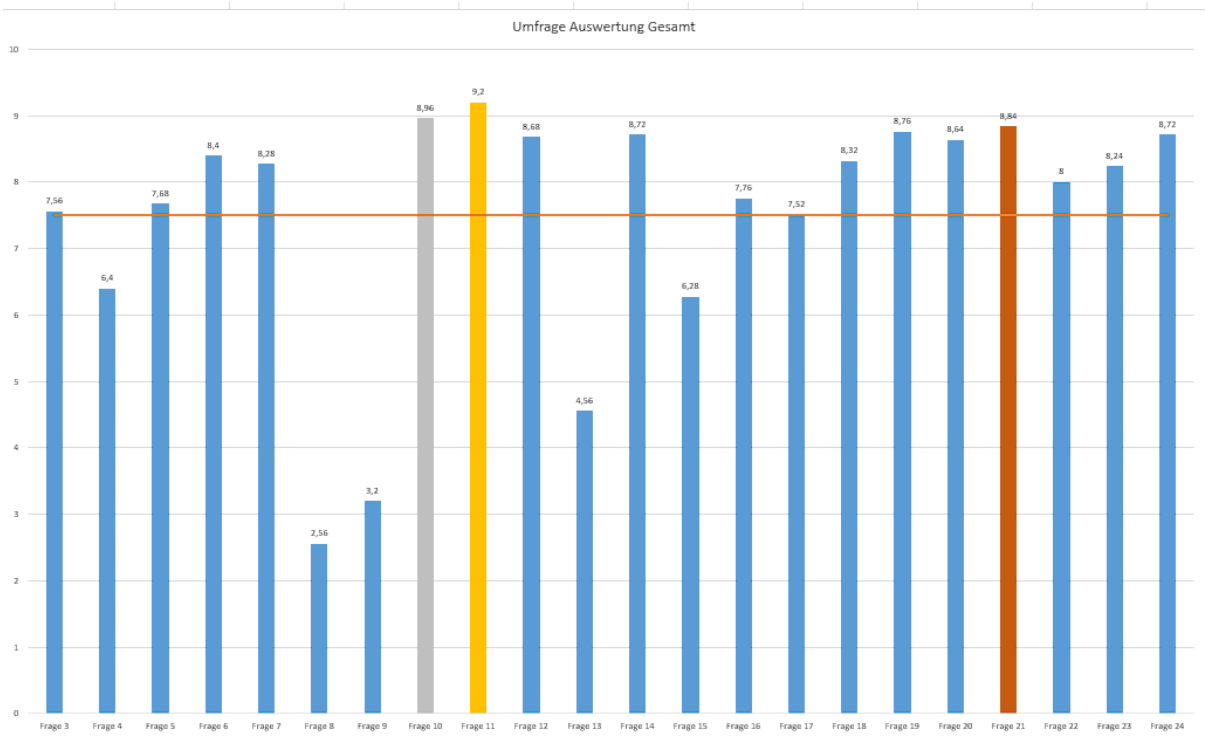


Abbildung 17 Gesamtauswertung

Auch finden sich hier die Mittelwerte aller anderer Fragen, mit Ausnahme von Frage 1 und Frage 2. Auch wurde in das Diagramm eine Mittelwertlinie eingefügt. Diese kann als eine Art Richtwert dienen. Auffallend ist hierbei, dass bis auf sechs Fragen jede Frage diese Linie überschreitet. Dies ist für diese Arbeit nicht relevant, allerdings ist es trotzdem ein interessantes Ergebnis.

6.5 Die drei meistgenannten Faktoren

Werden „*Abbildung 18*“ bis „*Abbildung 19“* betrachtet, zeigt sich, dass die Differenz zwischen den Mittelwerten dieser drei Fragen nicht sehr groß ist. Dies ist vor allem bei den Fragen 10 (Kommunikationsfluss im Unternehmen) und 21 (Definition of Ready, Definition of Done) der Fall. Der Mittelwert von Frage 10 und der Mittelwert von Frage 21 unterscheiden sich lediglich um eine Differenz von „*0,08“.*

Somit zeigt sich schnell, dass diese drei Faktoren ähnlich wichtig sind und sich vermutlich auch gegenseitig beeinflussen und die Grenzen zueinander auch oft nicht klar trennbar sind. Dies trifft aber auf alle hier genannten Faktoren zu. Die Faktoren beeinflussen sich nicht nur gegenseitig, sondern haben auch eine gewisse Abhängigkeit zueinander, denn der Erfolg eines Faktors kann durchaus von dem Erfolg eines anderen Faktors abhängig sein. Dies zeigt sich vor allem schnell, wenn als Beispiel die drei meistgenannten Faktoren betrachtet werden, denn unklare Prozesse benötigen erhöhte Kommunikation in Form von Rückfragen an den Auftraggeber. Durch diesen erhöhten Bedarf an Kommunikation entsteht mehr Wartezeit und die Produktivität der Mitarbeiter sinkt. Ist zusätzlich noch die Kommunikation in einem Unternehmen schlecht, erhöht sich die Wartezeit weiter. Durch dieses kurze Beispiel sollte aber die bereits erwähnte Abhängigkeit zwischen den Faktoren gezeigt werden.

Diese gegenseitige Beeinflussung kann allerdings auch positiv sein. Sind die Prozesse gut ausgearbeitet, ist weniger Kommunikation nötig. Somit steigt die Produktivität der Mitarbeiter und auch die Erfolgschance des Teams und des Projektes. Auch senkt eine gute Kommunikation Wartezeiten und somit steigt auch die Produktivität. Deshalb sollte es im Interesse der Betroffenen sein, dass Schwachpunkte im Unternehmen so bald wie möglich überarbeitet werden. Denn je produktiver jeder einzelne Mitarbeiter ist und sein kann, umso mehr Arbeit kann auch verrichtet werden.

6.5.1 Frage 11 Einfache und Verständliche Arbeitsprozesse

Mit einem Mittelwert von 9,2 ist diese Frage die mit dem höchsten Ergebnis, und somit auch der häufigste genannte Faktor. Wird „*Abbildung 18*“ betrachtet, ist sehr klar ersichtlich, dass hier eine recht eindeutige Tendenz der Antworten vorliegt. Lediglich eine Antwort senkt hier den Mittelwert. Hätte diese Person wie die meisten Befragten, mit einer 10 geantwortet, wäre der Mittelwert bei einer 9,4 und die Standardabweichung bei 0,7. Aber bereits jetzt hat diese Frage eine der geringsten Standardabweichungen. Dies bedeutet also, dass es äußerst wichtig für den Erfolg des Projektes und des Teams ist, dass die verwendeten Arbeitsprozesse klar, verständlich und auch einfach sind.

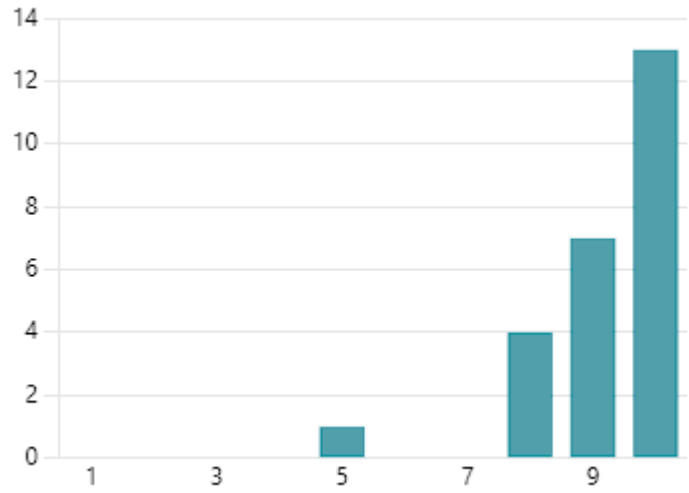


Abbildung 18 Frage 11

Unklare Prozesse führen eher dazu, dass Rückfragen von den Betroffenen gestellt werden müssen, da einige wichtige Punkte fehlen oder unklar sind. Dies führt dazu das langsamer gearbeitet wird und die Produktivität sinkt. Durch unklare Prozesse erhöht sich auch die Fehlerquote, da Prozesse nicht genau definiert sind, auch sind die notwendigen Arbeitsschritte unklar und somit wird auch falsch gearbeitet. Mehr Fehler heißt auch mehr Korrektur und Nacharbeit in einem späteren Arbeitsschritt. Somit wird die Arbeitsleistung des Teams weiter reduziert. Auch wird so eine strengere Qualitätskontrolle eingeführt, welches wieder mehr Arbeitsschritte benötigt, und den Output eines Teams noch weiter verringert.

Dies trifft auch zu, wenn Arbeitsprozesse zu komplex sind. Zu komplexe Prozesse führen zu einer langsameren Abarbeitung der Prozesse. Weiterhin steigt durch diese Komplexität auch die Chance

auf Fehler. Dies bedeutet für das Team wieder vermehrte Rückfragen und Nachbearbeitung. Eine weitere Folge von komplexen Prozessen ist, dass neue Mitarbeiter eine längere Einschulung benötigen, da diese sich in komplexe Prozesse einarbeiten müssen. Komplexe Prozesse benötigen auch eine ausführlichere Dokumentation. Diese muss auch regelmäßig überarbeitet werden, da sonst mehr Verwirrung entsteht. Ein weiterer Aspekt ist, dass, wenn ein Prozess überarbeitet wird, diese Überarbeitung auch leichter fällt, wenn dieser einfach, klar und gut dokumentiert ist.

Einfache Prozesse allerdings ermöglichen ein schnelleres Arbeiten, da hier keine oder weniger Zeit gebraucht wird für Rückfragen oder das Lesen von umständlicher Dokumentation. Auch verringern einfache Prozesse die Chance auf Fehler, da diese Prozesse eher verständlich und leichter zu lernen sind. Dasselbe trifft auch auf verständliche Prozesse, denn diese benötigen ebenfalls nur wenig Dokumentation. Ebenfalls ist die Zeit, welche Mitarbeiter benötigen, sich in diese neuen Prozesse einzuarbeiten, deutlich kürzer. Ein weiterer positiver Aspekt ist ebenfalls, dass das Optimieren und Adaptieren dieser einfachen und verständlichen Prozesse deutlich einfacher und kostensparender sind. Je mehr Prozesse verwendet werden, umso mehr ist auch deren Effekt spürbar, sei es positiv oder negativ. Somit sollte es ein großes Anliegen eines Unternehmens und Teams sein, die benötigten Arbeitsprozesse und Schritte so einfach und gut verständlich wie möglich zu gestalten. Auch sollten diese Prozesse in regelmäßigen Abständen adaptiert werden, um so eine noch bessere Produktivität zu bekommen und den eigenen Erfolg zu maximieren.

Auf eine genaue Definition, wann ein Prozess als einfach und verständlich gilt, kann hier leider nicht eingegangen werden, da dies von vielen unterschiedlichen Faktoren abhängig ist. Allerdings lässt sich dies schnell in einem Dialog mit den Prozessnutzern ausfindig machen, da diese ihre Arbeitsprozesse kennen und somit auch eventuelle Schwachstellen aufzeigen können. Ebenfalls kann der positive Aspekt leicht gemessen werden. Dies kann durch eine Vielzahl an unterschiedlichen Indikatoren geschehen.

.

6.5.2 Frage 10 Kommunikationsfluss des Unternehmens

Diese Frage ist mit einem Ergebnis von 8,96 der zweit meistgenannte Faktor. Auch bei dieser Frage zeigt sich, wenn „*Abbildung 19*Abbildung 18“ betrachtet wird, dass die Streuung der Antworten gering ist. Lediglich eine Person empfindet diesen Faktor als eher unwichtig, also findet sich auch hier ein „*Ausreißer*“, welcher sich ebenfalls in Frage 11 findet.

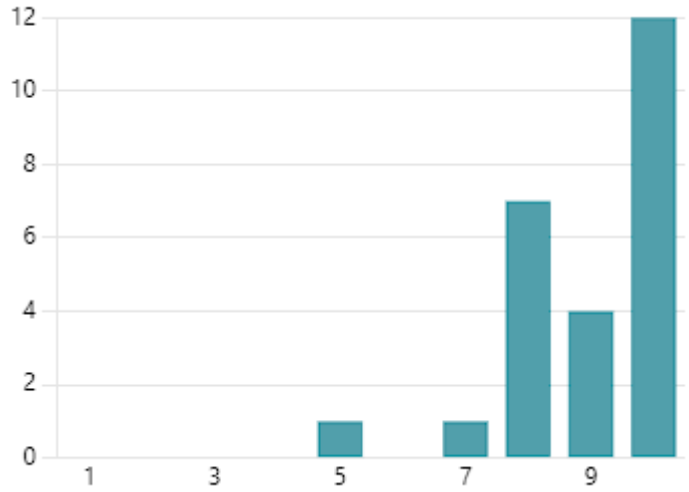


Abbildung 19 Frage 10

Dies bedeutet also, dass eine gute Kommunikation in einem Unternehmen einen großen Einfluss auf den Erfolg eines Projektes hat. Der Grund dafür ist recht schnell gefunden, denn wenn die Kommunikation in einem Unternehmen langsam oder unklar ist, trifft die benötigte Information später am Ziel ein. Umso schneller Informationen aufgenommen und weitergegeben werden, umso schneller kann auch eine entsprechende Aktion beziehungsweise Reaktion gesetzt werden. Allerdings muss die Information auch mit dem richtigen Inhalt weitergegeben werden. Es ist leider nicht förderlich, wenn eine Information zwar schnell am Ziel ist, aber mit dem falschen Inhalt.

Diese Verzögerung in der Kommunikation kann dazu führen, dass Dinge umgesetzt werden, welche nicht mehr den Bedürfnissen des Kunden entsprechen. Auch können Informationen durch eine schlechte Kommunikation verfälscht werden. Diese Verfälschung der Information ist problematisch, da so benötigte Features falsch oder mangelhaft implementiert werden. Hier muss man nur an das Beispiel des Kinderspiels „Stille Post“ denken, wo bereits einfache Sätze durch die lange Kommunikation verfälscht werden. Ein weiterer Problemfaktor ist, dass, wenn Zuständigkeiten nicht genau definiert sind, es dauert, bis die richtige Kontaktperson gefunden ist, und bis diese gefunden ist, geht wichtige Zeit verloren. Oder es muss erst eine lange Kommunikationskette durchlaufen werden, bis die Information am Ziel angelangt. Je größer das

Unternehmen ist, umso mehr Kommunikationsstellen hat dieses, und umso stärker sind auch schlechte Einflüsse spürbar, da diese Wartezeiten sich summieren.

Wird allerdings gute Kommunikation betrachtet, ist es nicht verwunderlich, dass dieser Faktor so einen hohen Wert erreicht hat. Kurze Kommunikationswege verringern auch die möglichen Wartezeiten auf Antwort. Die Information kommt schnell und auch unverfälscht am Ziel an. So kann die Information schnell verarbeitet werden und eine entsprechende Aktion gestartet werden. Auch helfen diese kurzen Kommunikationswege dabei, dass Information weniger verloren geht. Da im Optimalfall eine direkte Kommunikation mit der Ansprechperson erfolgen kann, ohne dass weitere Stellen durchlaufen werden müssen.

Dies zeigt klar, dass Arbeitnehmer und Arbeitgeber einen Fokus darauflegen sollten, dass ihre Kommunikationswege und Antwortzeiten kurz sind. Dies betrifft die Kommunikation in einem Team, in einer Abteilung und weiterhin in einem ganzen Unternehmen. Auch darf hierbei die Kommunikation mit und zu den Kunden nicht vernachlässigt werden. Es ist nicht förderlich, wenn die interne Kommunikation hervorragend ist, aber ein Kunde keine Antwort bekommt. Es müssen hier also mehrere Wege der Kommunikation betrachtet werden. Dies ist auch unabhängig von dem Medium der Kommunikation. Es sollte egal sein, ob eine Anfrage per E-Mail, per Telefon oder Ähnliches eingelangt wird. Die Antwort sollte schnell, präzise, aber auch klar sein, und es sollte dem Kunden auch ein direkter Kontakt gegeben werden, an den er sich wenden kann, wenn weiterer Bedarf besteht.

Diese Optimierung muss aber individuell betrachtet werden. Deshalb findet sich in vielen agilen Methodiken Kommunikation als einer der wichtigsten Punkte, da mithilfe von Kommunikation die Produktivität eines Teams stark beeinflusst werden kann. Wie bereits bei den Prozessen genannt, sollten die Kommunikationswege regelmäßig kontrolliert und, wenn nötig, überarbeitet werden.

6.5.3 Frage 21 Definition of Ready, Definition of Done

Diese Frage ist mit einem Mittelwert von 8,84 der dritt meistgenannte Faktor. Auch hier ist die Streuung der Antworten sehr gering sie beträgt 1,17 lediglich eine Antwort hat den schnitt gesenkt, dies wird auch offensichtlich, wenn „*Abbildung* 19“ betrachtet wird.

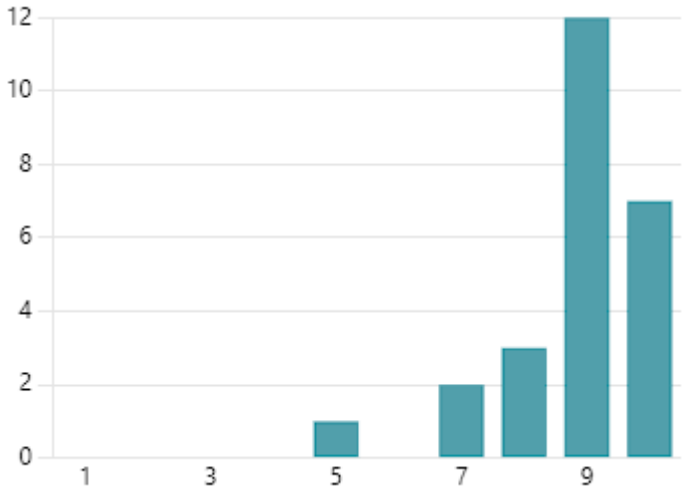


Abbildung 19 Frage 21

Die Gründe, wieso dieser Faktor wichtig ist, sind nach kurzer Überlegung auch gefunden. Wenn die Definitionen für „*Definition of Ready*“ und „*Definition of Done*“ aus Kapitel „*2.3.2.1*“ noch einmal gelesen. Die gute Ausformulierung dieser Definitionen hilft stark dabei, den Arbeitsfluss zu beschleunigen und zu vereinfachen, und sie dienen ebenfalls dem Vermeiden von unnötiger Kommunikation, welche einfach vermieden werden könnte.

Ist die „*Definition of Ready*“ gut definiert, sind vorab alle wichtigen Punkte geklärt, die ein Entwickler benötigt, um die Entwicklung zu starten. Hierbei kann es sich zum Beispiel handeln:

- Funktion
- verwendete Daten

Diese klare Definition benötigt somit weniger Rücksprachen mit dem Kunden beziehungsweise Auftraggeber, denn bei Rückfragen sollte es sich um Ausnahmen handeln und nicht um die Regel.

Ist die „*Definition of Done*“ gut definiert, hilft es dem Kunden oder dem Product-Owner bei der Endabnahme. Auch hier wird das Benötigen von Rücksprachen verringert. Dies kann durch das Definieren von sogenannten „*Acceptance Criteria/Abnahmekriterien*“ erfolgen. Durch die

geringeren Rücksprachen ist mehr Zeit für neue Tickets oder Ähnliches, und der Arbeitsfluss wird weniger oft unterbrochen.

Ist allerdings keine der beiden Definitionen gut getroffen, werden oft Rücksprachen gehalten, da Arbeitsrelevante Information für die betroffene Person fehlen. Durch diese Rücksprachen verzögert sich auch der weitere Prozess, da auf eine Antwort gewartet wird.

Wird wieder der positive Aspekt dieser beiden Definitionen betrachtet, wird auch schnell klar, wieso diese so wichtig sind. Ist die „*Definition of Ready*“ in Verwendung, kann initial kontrolliert werden, ob die zu implementierende Funktion alle notwendigen Informationen besitzt, um abgearbeitet zu werden. Dies gewährleistet ein Minimum an Informationen, welches dem Entwickler dabei hilft, schneller und kontrollierter zu arbeiten und so seine Produktivität zu maximieren.

Ist der Entwickler mit der Implementierung fertig, gewährleistet die „*Definition of Done*“ dem Kunden oder der Person, welche die Funktionalität kontrolliert, dass alle notwendigen Informationen vorhanden sind. Dies hilft dieser Person ebenfalls, ihre Produktivität zu erhöhen.

Somit sollte offensichtlich sein, wieso Teams einen großen Wert auf diese beiden Definitionen legen sollten, auch wenn dies erfahrungsgemäß leider nicht sehr leicht ist. Denn diese Definition kann von Fall zu Fall unterschiedlich sein. Der Leitsatz sollte allerdings sein „*so grob wie möglich, so detailliert wie nötig*“. So wird versucht, diese beiden Definitionen so allgemein wie nötig zu halten. Es ist aber ratsam, individuell zu kontrollieren und gegebenenfalls anzupassen.

6.6 Interessante Ergebnisse

Nun möchte ich hier auf weitere interessante Ergebnisse eingehen. Ich habe versucht, eine eventuelle Begründung zu geben, wieso ein Faktor ein gewisses Ergebnis erreicht hat. Diese Vermutungen wurden getroffen aufgrund dessen, dass ich mich selbst viel mit diesem Thema beschäftigt habe und meiner Erfahrung in der agilen Softwareentwicklung. Diese Arbeit beschäftigt sich allerdings nicht mit der Untersuchung der Gründe, wieso ein Faktor ein gewisses Ergebnis erzielt hat, und deshalb möchte ich mich auch nicht zu sehr mit den eventuellen Gründen beschäftigen.

6.6.1 Frage 8 Altersgruppe

Mit einem Mittelwert von 2,56 ist dieser Faktor der wenig bedeutendste Faktor für die Teilnehmer der Umfrage. Wird „*Abbildung 20*“ betrachtet zeigt sich auch schnell, dass die befragten sich eher einig sind bei der Bewertung dieser Frage

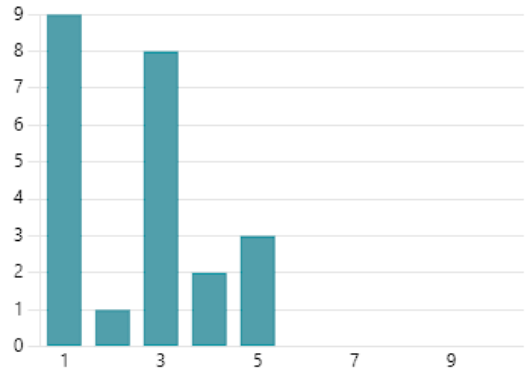


Abbildung 20 Frage 8

Dies kam überraschend, da ich diesem Faktor doch eine etwas höhere Bedeutung zugesprochen hätte, da es doch generationsabhängige Unterschiede gibt. Diese Unterschiede können vielseitig sein. So ist unter der jüngeren Generation Home-Office deutlich beliebter, während ältere Generationen eher eine Office-Präsenz bevorzugen. Ein weiterer Unterschied ist so auch die Form des gewünschten Feedbacks (Tolbize 2008). Diese Unterschiede können also auch zu Konflikten am Arbeitsplatz führen, da hier auch eine unterschiedliche Wahrnehmung stattfinden kann. Deshalb kam dieses Ergebnis für mich etwas unerwartet, da ich dachte, dass der Generationenunterschied doch eine deutlich höhere Gewichtung erhalten würde. Es ist allerdings sehr positiv, dass dieser Aspekt anscheinend für die Befragten einen sehr kleinen Einfluss hat.

6.6.2 Frage 9 Kultur

Mit einem Mittelwert von 3,2 ist dies der zweit unwichtigste Faktor, aber dafür der Faktor mit der höchsten Standardabweichung, welches ersichtlich wird, wenn „*Abbildung 21*„ betrachtet wird.

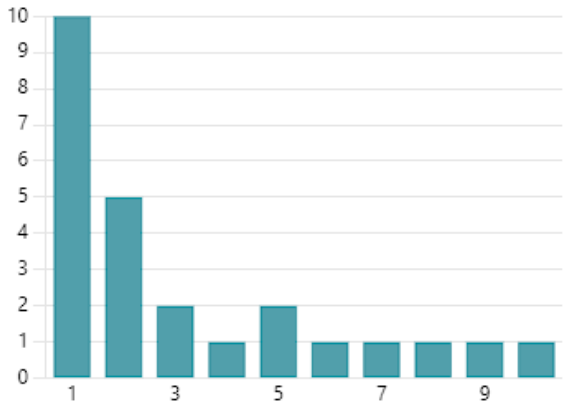


Abbildung 21 Frage 9

Dieses Thema scheint bei dem Befragten stark zu polarisieren, auch wenn die Tendenz deutlich für einen unwichtigen Faktor spricht. In internationalen Unternehmen kommt man früher oder später in den Kontakt mit Personen aus anderen Kulturen. Diese Kulturen können sich stark voneinander unterscheiden. Werden Mitarbeiter nicht im Umgang mit anderen Kulturen geschult, kann dies leider schnell zu Missverständnissen führen, aufgrund von kulturellen Unterschieden. So kann eine harmlose Geste in einer Kultur als Beleidigung in einer anderen aufgefasst werden. Als Beispiel kann das „*OK*“-Handzeichen in einigen Ländern als Beleidigung gesehen werden. (Anderson, et al. 2019)

Aber dieser Faktor dürfte nur eine geringe Rolle spielen. Dies kann sein, da Mitarbeiter öfter Schulungen erhalten, um diesen Konflikten vorzubeugen. Auch findet durch das tägliche Zusammenarbeiten ein natürlicher Kulturaustausch statt. In Zeiten der Digitalisierung geschieht dies auch schneller und häufiger als es vielleicht früher der Fall war. So kann sich auch einfach selbstständig informiert werden durch eine Internetrecherche. Auch erhalten Mitarbeiter, welche in internationalen Unternehmen arbeiten, oft Schulungen bezüglich anderer Kulturen und dazu, was in diesen Kulturen als höflich oder unhöflich angesehen wird. Diese Schulungen helfen dabei, Missverständnissen vorzubeugen.

6.6.3 Frage 7 Sprachkenntnisse

Mit einem Mittelwert von 8,28 ist diese Frage doch höher gewertet, auch ist hier die Streuung der Werte deutlich höher verglichen mit anderen hier gezeigten Faktoren, wenn „*Abbildung 22*“ betrachtet wird.

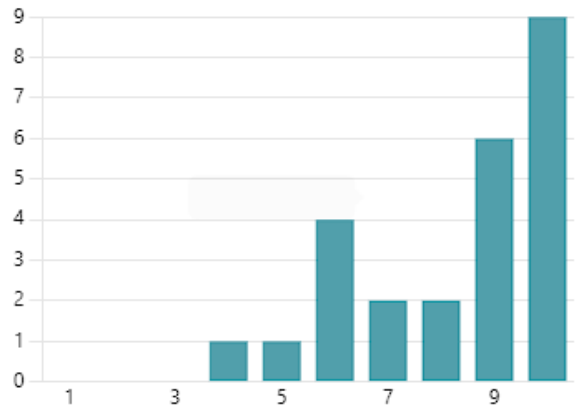


Abbildung 22 Frage 7

Dieses Ergebnis finde ich besonders interessant, vor allem wenn die Ergebnisse der „*Frage 8*“ und „*Frage 9*“ betrachtet werden. Denn die Faktoren Kultur und Alter waren anscheinend eher unwichtig. „*Frage 7*“ allerdings erhielt eine deutlich höhere Wertung als die beiden anderen Fragen.

Das ist wahrscheinlich auch leicht zu erklären, denn „*Frage 10*“ ist der zweitmeist genannte Faktor, und dieser ist die Kommunikation in einem Unternehmen. Diese zwei Faktoren gehen auch ineinander über. Denn Sprachen fließen auch direkt in die Kommunikation eines Unternehmens mit ein. Gerade in internationalen Unternehmen wird dies immer wichtiger, da Englisch oft die gemeinsame Sprache des Unternehmens ist. Wird diese gemeinsame Sprache allerdings nicht ausreichend beherrscht, ist die Kommunikation gestört. Ist die Kommunikation in einem Team oder Unternehmen aufgrund einer sprachlichen Barriere also schlecht, hindert dies das Arbeiten. So werden Dinge falsch umgesetzt oder gar nicht umgesetzt, aus dem einfachen Grund, dass einander nicht richtig oder gar nicht verstanden wird.

6.6.4 Frage 15 Agile Methodik

Diese Frage weist einen Mittelwert von 6,28 auf mit, einer doch merkbaren Streuung, was „*Abbildung 23*“ gut zeigt.

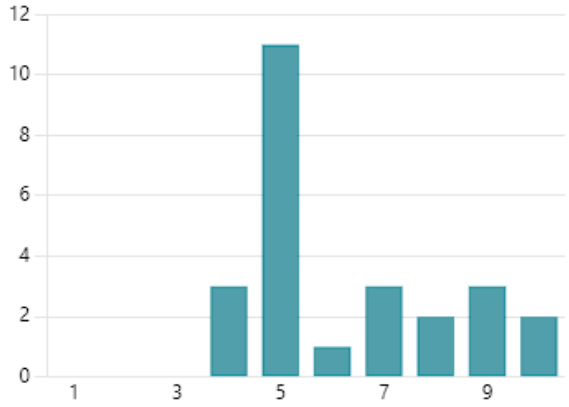


Abbildung 23 Frage 15

Diese Arbeit beschäftigt sich mit den Erfolgsfaktoren der agilen Methodik. Überraschend ist es also, dass dieser Faktor es nicht einmal über unseren erstellten Richtwert der Mittellinie geschafft hat. Wird „*Abbildung 17*“ betrachtet, ist ersichtlich, dass diesem Faktor eine geringe Bedeutung zugesprochen wird. Auch wenn in vorherigen Kapiteln immer wieder betont wurde, wie wichtig agiles Arbeiten ist. Vermutlich ist der Grund der, dass agile Methodiken oft nur verwendet werden, allerdings aber falsch verstanden oder angewendet werden. Auch kann es sein, dass andere Faktoren einen deutlich höheren Einfluss auf den Erfolg eines Projektes haben. Natürlich beinhaltet Agilität viele oben genannten Faktoren, wie gute Kommunikation oder ein gutes Verhältnis zu dem Kunden. Allerdings bedeutet das Verwenden von agilen Methodiken nicht gleich, dass diese beinhalteten Faktoren automatisch mit angewendet werden, wenn diese nicht aktiv verwendet werden. Aus diesem Grund vermute ich, dass die Verwendung einer agilen Methodik nicht so wichtig ist wie einige andere der hier genannten Faktoren. Agilität hilft nicht bei komplex definierten Arbeitsprozessen, welche abgearbeitet werden müssen, oder bei mangelnder oder schlechterer interner Kommunikation. Sie soll lediglich dem Team helfen, den Arbeitsaufwand besser zu meistern. Allerdings ist es interessant, dass, obwohl Agilität generell als wichtig und doch oft als „*best practice*“ angesehen wird, es doch als eher unwichtiger Faktor erachtet werden kann. Es ist allerdings wahrscheinlich auch so, dass schlechte Agilität nicht zwingend besser ist als zum Beispiel eine gute Verwendung des Wasserfall-Modells.

6.6.5 Frage 13 Self Managing Teams oder Management durch das Unternehmen

Diese Frage weist einen Mittelwert von 4,56 auf, mit einer hohen Streuung siehe „*Abbildung 24*“. Allerdings ist hier eine klare Tendenz zu Self Managing Team zu erkennen.

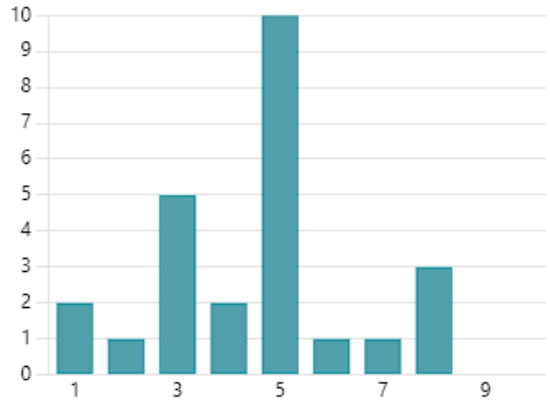


Abbildung 24 Frage 13

Werden die Agilen Werte und vor allem der Scrum-Guide betrachtet, zeigt sich auch, warum es diese Tendenz gibt. Denn kleine Teams kennen ihre Kapazitäten und wissen genau, wie viel Arbeit sie in einer gewissen Zeit (Sprint) erledigen können, wo es eventuelle Schwachstellen und Stärken im Team gibt. Auch ist hier oft die Kommunikation im Team deutlich schneller und effizienter als es in einer komplexen Unternehmensstruktur sein könnte. Auch optimieren kleinere Teams ihre internen Prozesse schneller und erhöhen somit auch ihre Produktivität, da die Stärken und Schwächen der anderen Mitglieder bekannt sind.

Dies kann nicht stattfinden, wenn ein starkes Mikromanagement durch das Unternehmen stattfindet, da dieses gar keine Nähe zu den Teams hat, aufgrund der unterschiedlichen Rolle und eventuellen örtlichen Distanz. Auch fehlt Management unter Umständen die Nähe zu der Materie, welche Teams besitzen. So können Manager eventuell gar nicht einschätzen, was für ein Aufwand ein Projekt oder eine Aufgabe ist. Allerdings ist hier wahrscheinlich eine Mischung aus beidem ein guter Ansatz, welcher auch das Ergebnis der Frage klar zeigt. Den auch die individuellen Teams brauchen ein generelles, beziehungsweise übergeordnetes Ziel, da die einzelnen Teams an unterschiedlichen Strängen ziehen und so eventuell das große Ziel aus den Augen verlieren können. Auch sollten einige große Entscheidungen durch ein Management getroffen werden. Dies würde eine Teilung der Arbeit bedeuten. So wird vom Unternehmen eine grobe Richtung und Richtlinien an die Teams gestellt, aber die Umsetzung sollte bei den Teams liegen.

Allerdings ist hier die Trennung schwer, da diese Arbeitsweise doch neu ist und für viel Unternehmensmanagement und Teammitglieder noch eher Neuland ist.

6.6.6 Frage 24 ausführliche Dokumentation

Mit einem Mittelwert von 8,72 ist dies einer die häufigen genannten Faktoren, allerdings lässt sich auch hier eine eher große Streuung der werte erkennen, wenn ein Blick auf „*Abbildung 25*“ geworfen wird.

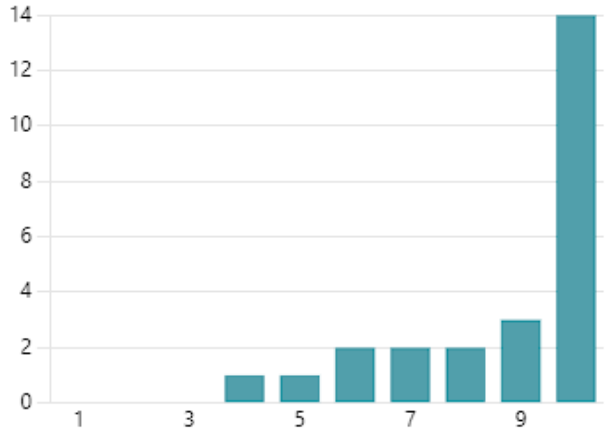


Abbildung 25 Frage 24

Auch wenn dieser Punkt doch eher gegen die Agilen werte spricht, denn laut diesem sollte eine funktionierende Software wichtiger sein als eine ausführliche Dokumentation. Aus diesem Grund finde ich es interessant, dass dieser Faktor so häufig genannt wurde.

Hier kann auch eine eventuelle Verbindung zu der Rolle der Befragten gezogen werden. Da es sich bei einem Großteil der Befragten um „*Business-Analysten*“ handelt, ist die Dokumentation sehr wichtig. Denn Business Analysten müssen die Geschäftslogik verstehen, mit der die Geschäftsprozesse durchgeführt werden. Ist keine oder mangelnde Dokumentation vorhanden, kann dies nicht vernünftig geschehen. Aufgrund der mangelhaften oder schlechten Dokumentation werden Prozesse falsch verstanden. Durch dieses falsche oder fehlende Verständnis werden Prozesse auch nicht korrekt an die Entwickler weitergegeben und der Prozess falsch implementiert.

Somit lässt sich ein Rückschluss ziehen, wieso der Faktor so häufig genannt wird, da dieser für Business-Analysten ein sehr wichtiger Punkt ist, der sie in ihrer täglichen Arbeit stören kann oder im besseren Fall ihre tägliche Arbeit sehr erleichtert. Daher sollte ein großer Wert darauf gelegt werden, dass Dokumentation zwar nicht der Fokus der Arbeit ist, allerdings auch nicht vernachlässigt wird. Da sonst, wie eben beschrieben, es zu häufig zu Rückfragen kommt oder zu einer falschen Implementierung der gewünschten Geschäftsfälle. Wahrscheinlich sind ein regelmäßiges Aktualisieren und Überarbeiten der Dokumentation auch ein valider Schritt, um hier die Qualität maßgeblich zu erhöhen.

7. Beantwortung der Forschungsfrage und Evaluierung der Hypothese

7.1 Beantwortete Forschungsfrage

Um ein Projekt abschließen zu können, benötigt es Erfolge, und diese Erfolge sind natürlich an gewisse Faktoren gebunden, allerdings lassen sich diese Faktoren nicht so einfach identifizieren. Aus diesem Grund hat sich folgende Forschungsfrage gebildet:

„Welche sind die 3 meistgenannten Erfolgsfaktoren für das agile Arbeiten in Softwareprojekten?“ “

Diese Forschungsfrage, konnte im Zuge dieser Arbeit folgend beantwortet werden: Die drei meistgenannten Erfolgsfaktoren für das agile Arbeiten in Softwareprojekten sind:

- Einfache und Verständliche Arbeitsprozesse
- Ein Klarer Kommunikationsfluss des Unternehmens
- Eine gute „*Definition of Done*“ und „”*Defintion of Ready*”

7.2 Evaluierung der Hypothese

Die in Kapitel „*1.2.2 Hypothese*” formulierte Hypothese lautete.

Die 3 meistgenannten Erfolgsfaktoren für das agile Arbeiten in Softwareprojekten sind:

- Klare Definition des Projektes/Scopes
- Ein Klarer Kommunikationsfluss innerhalb des Unternehmens
- Einfache und Verständliche Arbeitsprozesse

Wird die Hypothese jetzt mit der beantworteten Forschungsfrage verglichen, decken sich diese in zwei von drei Fällen. Denn sowohl in der beantworteten Forschungsfrage als auch in der Hypothese wurden die Punkte „*Ein klarer Kommunikationsfluss innerhalb des Unternehmens*“ sowie „*Einfache und verständliche Arbeitsprozesse*“ genannt. Auch wenn hier eine unterschiedliche Gewichtung beider Faktoren vorliegt.

In meiner Hypothese wurde allerdings noch der Erfolgsfaktor „*Klare Definition des Projektes/Scopes*“ genannt. Dieser war allerdings keiner der meistgenannten Faktoren und kommt deshalb auch nicht in der beantworteten Forschungsfrage vor. Somit hat sich die von mir formulierte Hypothese als falsch herausgestellt.

Es ist aber interessant, dass die Definition of Done und Ready höher gewichtet wurde als die Definition des Projektes Umfanges.

Dies ist aber wahrscheinlich darauf zurückzuführen, dass ein unklarer Projektumfang nicht so hinderlich ist für den Erfolg eines Projektes, als wenn täglich Rückfragen zwischen Teams gehalten werden müssen. Diese Rückfragen sind nötig aufgrund der fehlenden Definition von Abnahmekriterien sowie der Startkriterien für die Implementierung einer Anforderung. Ein weiterer Grund kann auch sein, dass in agilen Projekten sich der Umfang eines Projektes ständig ändert und somit eine gewisse Unschärfe herrscht, und aufgrund dessen kann gar kein klarer Scope abgesteckt sein. Oder im Falle von SCRUM, dass der Sprint so klein ist, dass hier eine kleine Änderung des Scopes irrelevant ist.

8. Zusammenfassung und Ausblick

Softwareentwicklung hat in ihrer Lebenszeit einige Änderungen durchlaufen, aber eine der wichtigsten ist wahrscheinlich die Einführung von Agilen Methodiken. Die Autoren des Agilen Manifests wollten mit dieser neuen Arbeitsweise alte Konstrukte aufbrechen und die Entscheidungen an die Entwickler und Teammitglieder geben. Ebenfalls war die Vereinfachung der Kommunikation ein großes Ziel sowie dynamischeres Arbeiten. Wird „*Abbildung 9*“ nochmals betrachtet, wird also klar, wieso Agile Methodiken so oft verwendet werden. Dynamische und flexible Prozesse, die von kleinen, spezialisierten Teams definiert und betreut werden, sowie das Entwickeln von funktionaler Software anstatt des mühsamen Schreibens von Dokumentationen.

In der Theorie klingen diese Ansätze sehr gut und vor allem logisch, allerdings ist dies nicht alles, was den Erfolg eines agilen Projektes ausmacht. Mit diesem Thema hat sich diese Arbeit beschäftigt. Da nicht alle Faktoren genannt werden konnten, wurden nur einige genannt, die sich in meinen Augen als wichtig herausstellen. Die Auswahl der Fragen erfolgte primär durch:

- meine Persönliche Erfahrung
- durcharbeiten von Literatur
- Gespräche mit Kollegen
- Gespräche mit Bekannten

Aufgrund dieser obigen Faktoren wurden die Fragen formuliert, die in meiner Umfrage verwendet wurden. Mit der weiteren Erfahrung, die ich, während des Schreibens dieser Arbeit erhalten habe, hätte ich einige Fragen anders formuliert beziehungsweise andere Fragen gewählt. Die von mir durchgeführte Umfrage ergab: Die drei meistgenannten Erfolgsfaktoren für das agile Arbeiten in Softwareprojekten sind:

- Einfache und Verständliche Arbeitsprozesse
- Ein guter Kommunikationsfluss des Unternehmens
- Eine gute „Definition of Done“und „”Defintion of Ready”

Somit lässt sich zusammenfassen, dass in einem Projekt ein großer Fokus auf interner und externer Kommunikation liegen sollte, ebenso auf einfachen Prozessen sowie auf der Verwendung von einer „*Definition of Done*“ und der „*Definition of Ready*“. Wie diese Punkte zu erfüllen sind, muss allerdings auf einer individuellen Basis erfüllt werden. Hier kann und soll auch keine Anleitung gegeben werden, wie diese erfüllt werden können. Sicher ist allerdings, dass diese drei Faktoren viel Arbeit und auch immer wiederkehrende Überarbeitung brauchen, um optimal zu funktionieren. Nur weil die Kommunikation in einer Abteilung initial gut ist, bedeutet das allerdings nicht, dass dies weiterhin so ist. Hier ist eine ständige Überarbeitung der Prozesse und Schritte nötig, da sich ein Erfolgsfaktor sonst in einen Störfaktor entwickelt, welcher das Arbeiten verhindert.

Interessant könnte das Ergebnis dieser Arbeit für Unternehmen oder Personen sein, welche in der agilen Softwareentwicklung arbeiten. Da diese täglich mit diesen Erfolgsfaktoren konfrontiert sind und von dem Ergebnis dieser Arbeiten profitieren können, da sie die gewonnenen Erkenntnisse in ihre Arbeitsweise einfließen lassen können. Hierbei kann die Größe des Teams oder des Unternehmens vernachlässigt werden, da die Erfolgsfaktoren sich hier nicht ändern sollten. Natürlich lassen diese Ergebnisse sich nicht auf alle Firmen und Teams gleich anwenden. So müssten individuelle Anpassungen getroffen werden, um so den maximalen Effekt zu erzielen. Da unterschiedliche Arbeitsumfelder auch unterschiedliche Bedürfnisse haben. Aber es bietet einen guten Überblick darüber, welchen Faktoren mehr Bedeutung geschenkt werden sollte und welchen doch vernachlässigt werden können oder einer geringeren Bedeutung zugesprochen werden kann.

Weiters könnte auch die gegenseitige Beeinflussung der einzelnen Faktoren untersucht werden. So könnten sich vielleicht einzelne Kernfaktoren finden lassen. Auch könnte diese Umfrage um weitere Fragen erweitert werden, um so ein weiteres Spektrum abzudecken. Ebenfalls könnten Schwerpunkte für einzelne Berufsgruppen gebildet werden. Natürlich wäre auch eine Spezialisierung für gewisse Branchen oder Bereiche der Softwareentwicklung möglich. All dies konnte in dieser Arbeit leider nicht überprüft werden.

Abschließend lässt sich allerdings festhalten, dass die verwendeten Arbeitsprozesse einen großen Einfluss auf den Erfolg von Projekten haben. Deshalb sollten diese regelmäßig kontrolliert, korrigiert oder bei Bedarf neugestaltet werden. Dasselbe trifft auch zu auf die Kommunikation in einem Unternehmen. Schwachstellen oder Schwierigkeiten sollten so schnell wie möglich erkannt und beseitigt werden, da sonst der Erfolg eines Projektes massiv gefährdet wird. Es sollte auch klar sein, dass je produktiver ein Team oder ein Unternehmen ist, umso mehr Umsatz kann es theoretisch generieren. Somit sollte es also in dem Interesse aller sein, dass hier alle Faktoren bedacht werden und so gut es geht, umgesetzt werden. So lassen sich leicht Kapitalsteigerungen erzielen. Nehmen wir als Beispiel: Ein Mitarbeiter braucht für einen optimierten Prozess 1 Minute und 20 Sekunden. Dies kann zum Beispiel sein, bevor Code in ein GitHub-Repository gepusht werden darf, muss ein Eintrag in eine Excel-File gesetzt werden. Wird dies jetzt mit der Anzahl der Mitarbeiter multipliziert, zeigt sich schnell, dass hier die verlorene Arbeitszeit stark wächst, wichtige Zeit, in der produktiv gearbeitet werden könnte. Hier handelt es sich jetzt nur um ein kleines Beispiel. Es soll einfach als kleiner Anstoß dienen.

9. Formeln, Fragenkatalog und Rohdaten

9.1 Formeln

Anbei die von mir verwendeten Formeln, welche angewendet wurden für das Errechnen der Kennzahlen. Als Beispiel wurden die angewendeten Formeln für Frage 3 genommen, bei den verwendeten Formeln handelt es sich allerdings bei jeder Frage um dieselbe verwendete Formel.

Mittelwert

=AVERAGE(Table1[Frage 3])

Standardabweichung

=STDEV.S(Table1[Frage 3])

Modal

=MODE.MULT(Table1[Frage 3])

Median

=MEDIAN(Table1[Frage 3])

9.2 Fragenkatalog

Anbei der Fragenkatalog welcher verwendet wurde in dieser Arbeit. Die möglichen Antworten wurden nicht angehängt da es sich hier um dieselben möglichen Antworten hält.

1. Arbeiten Sie zurzeit, bzw. haben Sie bereits in einem Agilen Softwareentwicklung Umfeld gearbeitet
2. In was für einer Position sind bzw. Waren sie tätig?
3. Für wie wichtig halten Sie es, dass eine maximale Teamgröße nicht überschritten wird?
4. Für wie wichtig halten Sie das Abhalten von täglichen Meetings (Daily)?
5. Für wie wichtig halten Sie das regelmäßige Abhalten von Team Events?
6. Für wie wichtig halten Sie es, dass es ein gutes Kunden - Team Verhältnis gibt?
7. Für wie wichtig halten Sie es, dass alle Mitarbeiter dieselben Sprachkenntnisse besitzen? (Englisch, Deutsch, etc.)
8. Für wie wichtig halten Sie es, dass alle Mitarbeiter derselben Altersgruppe angehören?
9. Für wie wichtig halten Sie es, dass alle Mitarbeiter derselben Kultur angehören?
10. Für wie wichtig halten Sie einen guten Kommunikationsfluss des Unternehmens
11. Für wie wichtig halten Sie einfache und Verständliche Arbeitsprozesse

12. Für wie wichtig halten sie der Kommittent der Mitarbeiter?
13. Was halten sie für wichtiger Self Managing Teams oder Management durch das Unternehmen
14. Für wie wichtig halten Sie die klare Definition des Projektes Umfanges?
15. Für wie wichtig halten Sie es, dass eine agile Methodik verwendet wird?
16. Für wie wichtig halten Sie eine flache Organisation Hierarchie?
17. Wie sehr beeinflusst die Größe des Unternehmens das Agile Arbeiten?
18. Für wie wichtig halten sie Management Kommittent?
19. Für wie wichtig halten sie das Verwenden von Coding Standards?
20. Für wie wichtig halten sie es, aktuelle Technologien zu verwenden? (Programmiersprachen, Frameworks)
21. Für wie wichtig halten Sie eine gute Definition of Ready, Definition of Done?
22. Für wie wichtig halten Sie das regelmäßige bearbeiten (Refactoring) des Codes
23. Für wie wichtig halten Sie die regelmäßige Aktualisierung und Priorisierung des Backlogs durch den Product Owner?
24. Für wie wichtig halten Sie ausführliche Dokumentation?

9.3 Rohdaten

ID	In was für eine	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10	Frage 11	Frage 12	Frage 13	Frage 14	Frage 15	Frage 16	Frage 17	Frage 18	Frage 19	Frage 20	Frage 21	Frage 22	Frage 23	Frage 24
2	Scrum Master	8	10	9	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	Tester	2	4	4	5	6	3	3	9	9	7	5	7	5	3	9	4	7	4	7	4	7	4
4	Business Analyst	7	2	10	10	10	3	4	10	10	8	8	8	9	7	8	10	10	9	10	3	8	7
7	Scrum Master	10	7	4	8	8	3	3	8	10	10	1	8	8	8	8	8	8	8	8	10	8	6
9	Business Analyst	7	9	4	9	7	3	7	8	8	8	3	9	7	8	8	7	8	9	9	10	8	10
10	Product Owner	5	7	8	10	9	5	1	8	10	5	8	10	10	8	6	5	10	10	10	8	9	10
23	Entwickler	10	5	10	8	10	3	6	10	9	9	4	7	6	6	8	10	8	8	10	8	8	9
24	Business Analyst	10	7	9	10	9	5	9	8	8	9	1	10	7	8	8	9	10	8	9	7	8	10
25	Product Owner	10	5	7	9	10	1	1	10	10	10	5	10	5	10	10	10	10	10	9	8	9	10
35	Entwickler	7	10	8	10	10	2	2	10	10	10	3	6	10	10	5	8	10	10	10	10	8	8
36	Business Analyst	9	9	9	9	10	5	5	9	10	10	5	10	8	6	4	10	10	10	9	10	9	7
41	Tester	6	3	7	7	9	3	2	8	8	8	7	9	7	7	7	6	8	7	8	7	9	10
44	Business Analyst	7	5	10	9	10	1	1	8	9	5	3	7	5	10	8	5	9	8	7	8	7	6
45	andere Position	8	4	8	10	4	4	1	10	8	8	4	9	4	8	6	9	8	7	8	6	10	10
46	Scrum Master	10	7	10	7	7	4	1	7	10	10	5	10	5	8	8	10	10	10	10	10	8	10
49	Product Owner	7	3	7	8	6	1	1	10	9	9	6	6	9	8	3	10	5	9	9	5	10	8
51	Business Analyst	7	7	8	5	10	3	1	10	10	10	5	10	5	8	8	10	10	8	10	10	8	10
53	Product Owner	8	9	7	9	6	1	8	10	9	10	8	9	9	9	9	7	9	10	9	9	9	10
54	Business Analyst	7	7	7	9	6	3	1	10	10	10	5	10	5	8	8	10	10	10	9	10	8	10
55	Scrum Master	7	7	7	9	10	1	1	10	10	10	5	10	5	8	8	10	10	10	9	7	7	10
56	Tester	7	7	7	9	10	1	1	10	10	10	5	10	5	8	8	10	10	8	9	10	8	10
57	Product Owner	8	6	8	9	9	3	2	8	9	9	3	10	5	8	9	8	9	9	10	8	9	10
58	Tester	8	9	7	10	9	1	10	9	10	9	5	10	4	8	9	9	8	10	9	8	9	10
59	Entwickler	8	5	8	8	8	1	2	9	10	9	2	9	4	9	9	9	9	10	9	9	9	9
60	andere Position	6	6	9	8	9	1	2	10	9	9	3	9	5	8	9	9	8	9	9	10	8	9
	Modal	7	7	7	9	10	3	1	10	10	10	5	10	5	8	8	10	10	10	9	10	8	10
	Median	7	7	8	9	9	3	2	9	10	9	5	9	5	8	8	9	9	9	9	8	8	10
	Average	7,56	6,4	7,68	8,4	8,28	2,56	3,2	8,96	9,2	8,68	4,56	8,72	6,28	7,76	7,52	8,32	8,76	8,64	8,84	8	8,24	8,72
	Standard Deviati	1,804623691	2,217355783	1,749285568	1,554563176	1,860107524	1,386842938	2,798809271	1,27410099	1,154700538	1,62583312	1,938212235	1,541644144	1,947648154	1,535143859	1,734935157	1,95192213	1,479864859	1,604161255	1,178982612	2,081665999	1,051982256	1,83757086

Abbildung 27 Rohdaten

10. Abbildungsverzeichnis

Abbildung 1 Festplatten Preis Pro GB Entwicklung (Brandt 2014) 6
Abbildung 2 Software-Lebenszyklus (eigene Erstellung) 7
Abbildung 3 Einfaches Wasserfall Modell (Royce 1970) 14
Abbildung 4 Wasserfallmodell (Royce 1970) 15
Abbildung 5 Scrum Sprints (Henderson 2023) 20
Abbildung 6 Kanban Beispiel Board (eigene Erstellung) 21
Abbildung 7 Faktoren (eigene Erstellung) 23
Abbildung 8 Störfaktoren (Chow and Cao 2007) 24
Abbildung 9 Softwareentwicklungsmethoden Verwendung (Statista 2022) 26
Abbildung 10 Agile Methodiken in nicht IT-Abteilungen (Ravikumar 2023) 27
Abbildung 11 Verwendung Agile Methodik (Ravikumar 2023) 28
Abbildung 12 Anzahl der Publikationen (Torgeir, et al. 2012) 29
Abbildung 13 Erfolgsrate Wasserfall Methodik (Zippia 2022) 30
Abbildung 14 Erfolgsrate Agile Methodik (Zippia 2022) 30
Abbildung 15 Teilnahmefrage 38
Abbildung 16 Rollenfrage 38
Abbildung 17 Gesamtauswertung 39
Abbildung 18 Frage 11 41
Abbildung 19 Frage 21 45
Abbildung 20 Frage 8 47
Abbildung 21 Frage 9 48
Abbildung 22 Frage 7 49
Abbildung 23 Frage 15 50
Abbildung 24 Frage 13 51
Abbildung 25 Frage 24 52

11. Literaturverzeichnis

Altuwaijri, Fahad, und Maria Angela Ferrario. 2022. „Factors affecting Agile adoption: An industry research study of the mobile app sector in Saudi Arabia.“ *Journal of Systems and Software Volume 190*, 08.

Balzert, H. 2011. In *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb.*, von Der Software-Lebenszyklus, 109-110. Spektrum Akademischer Verlag.

Benning, Valerie. 2019. *scribbr.* 05. 04. Zugriff am 10. 10 2023. https://www.scribbr.at/statistik-at/korrelation/.

Brandt, Mathias. 2014. „Dramatischer Preisverfall bei Festplattenspeichern.“ *Statista.* 07. 08. Zugriff am 06. 10 2023. https://de.statista.com/infografik/2544/entwicklung-preis-pro-gigabyte-festplattenspeicher/.

Burkhard, Renz. 2012. „Vorgehen im Softwareentwicklungsprozess.“ Gießen: Technische Hochschule Mittelhessen, 12. 04.

Chow, Tsun, und Dac-Buu Cao. 2007. „https://www.researchgate.net/.“ *researchgate.* 26. 08. Zugriff am 23. 10 2023. researchgate.

Corona, Erika, und Filippo Eros Pani. 2023. „A Review of Lean-Kanban Approaches in the Software Development .“ *Department of Electrical and Electronic Engineering* , 01: 1-13.

Eby, Kate. 2019. *smartsheet.* 16. 07. Zugriff am 12. 10 2023. https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto.

Eichtstätt-Ingolstadt, Katholische Universtitä. kein Datum. *https://eo-vmw-jwpa.ku.de/.* Zugriff am 09. 10 2023. https://eo-vmw-jwpa.ku.de/journalistik/methoden/untersuchungsvorbereitung/verifikation-falsifikation/.

Fladerer, Anna. 2023. *bachelorprint.* 23. 03. Zugriff am 10. 10 2023. https://www.bachelorprint.at/methodik/umfrage-auswerten/.

Flandorfer, Priska. 2019. *scribbr.* 01. 04. Zugriff am 10. 10 2023. https://www.scribbr.at/statistik-at/regressionsanalyse/.

Franziska, Pfeiffer. 2018. *Sribbr.* 21. 01. Zugriff am 09. 10 2023. https://www.scribbr.de/methodik/fragebogen-erstellen/.

Gorans, Paul, und Philippe Kruchten. 2014. *A Guide to Critical Success Factors in Agile Delivery.* IBM Center for The Business of Government.

Henderson, Luke. 2023. „How To Manage A Sprint Cycle More Effectively.“ *niftypm.* 10. 08. Zugriff am 07. 10 2023. https://niftypm.com/blog/manage-sprint-cycle/#6_Include_inputs_from_previous_sprints.

Kent, Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, et al. 2001. *Agile Manifesto.* Zugriff am 25. 09 2023. http://agilemanifesto.org/.

Leipzig, Universität. kein Datum. *Universität Leipzig.* Zugriff am 02. 10 2023. https://home.uni-leipzig.de/methodenportal/qualivsquanti/.

Potvin, Rachel, und Josh Levenberg. 2016. „Communications of the ACM, vol. 59.“ *Communications of the ACM, vol. 59*, 78-87.

Ravikumar, Patel. 2023. *radixweb.* 10. 07. Zugriff am 03. 10 2023. https://radixweb.com/blog/agile-statistics#Methodology.

Royce, Winston.W. 1970. „Managing the Development of Large Software Systems.“ *Proceedings of IEEE WESCON*, 328-388.

Schikuta, Erich, Michael Derntl, und Helmut Wanek. 2019. „Grundlagen des Software Engineerings.“ In *Grundlagen des Software Engineerings*, von Erich Schikuta, Michael Derntl und Helmut Wanek, 8-10. Wiener Neustadt: Ferdinand Porsche Fernfachhochschule GmbH.

Schwaber, Ken, und Jeff Sutherland. kein Datum. *Scrum.* Zugriff am 25. 09 2023. https://www.scrum.org/.

ScrumEvent. kein Datum. *Scrum-Events: SCRUM Zertifizierung & SCRUM Schulung.* Zugriff am 07. 10 2023. https://www.scrum-events.de/scrum-antworten/was-ist-die-definition-of-ready-dor.

Sutherland, Jeff, und Ken Schwaber. 2020. *The Scrum Guide.* Ken Schwaber & Jeff Sutherland.

Team, Adobe Communications. 2022. *business.adobe.com.* 18. 03. Zugriff am 12. 10 2023. https://business.adobe.com/blog/basics/agile-manifesto.

Torgeir, Dingsøyr, Nerur Sridhar, Balijepally Gopal Venu, und Moe Brede Nils. 2012. „A decade of agile methodologies: Towards explaining agile software development,.“ *Journal of Systems and Software Volume 85, Issue 6*, 1213-1221. Zugriff am 06. 10 2023. https://www.sciencedirect.com/science/article/pii/S0164121212000532.

Tuckman, B. W. 1965. „Developmental sequence in small groups.“ *Psychological Bulletin* 63 (6): 384-399. Zugriff am 25. 09 2023. doi: doi:10.1037/h0022100.

Vailshery, Lionel sujay. 2022. „Statista.“ *Breakdown of software development methodologies practiced worldwide in 2022.* 15. 09. Zugriff am 03. 10 2023. https://www.statista.com/statistics/1233917/software-development-methodologies-practiced/.

Vocabulary.com´Dictionary. kein Datum. *"iteration".* Zugriff am 01. 10 2023. https://www.vocabulary.com/dictionary/iteration.

wikious. kein Datum. „Kanban (Development).“ *wikious.com.* Zugriff am 09. 10 2023. https://wikious.com/en/Kanban_(development).

Yngve, Lindsjørna, Gunnar R. Bergersena, Torgeir Dingsøyr, und Dag I.K. Sjøberga. 2016. „Teamwork quality and project success in software development:.“ *The Journal of Systems and Software 122* 274-286.

Zippia. 2022. *16 Amazing Agile Statistics [2023] ": What Companies Use Agile Methodology.* 07. 11. Zugriff am 03. 10 2023. https://www.zippia.com/advice/agile-statistics/.